



Application of Parabolic Equation methods  
to simulate in-ice Radio Wave Propagation  
for a subsurface exploration of Enceladus' ice crust using Borehole  
Ground-Penetrating Radar

**Bachelor Thesis**

zur Erlangung des akademischen Grades  
*Bachelor of Science im Fach Physik*

der Fakultät Mathematik und Naturwissenschaften  
Fachgruppe Physik  
der Bergischen Universität Wuppertal

vorgelegt von

**Gianluca Boccarella**  
(geboren in Wuppertal)

September 2021

**1. Gutachter:**

Prof. Dr. Klaus Helbing  
Bergische Universität Wuppertal

**2. Gutachter:**

Prof. Dr. Christopher Wiebusch  
RWTH Aachen

## Contents

<b>List of Figures</b>	<b>ii</b>
<b>List of Tables</b>	<b>iv</b>
<b>Acronyms</b>	<b>iv</b>
<b>Abstract</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Enceladus</b>	<b>2</b>
2.1 Structure . . . . .	3
2.2 Cryovolcanism and Surface Composition . . . . .	3
2.3 EnEx - Enceladus Explorer . . . . .	7
<b>3 Radioglaciology - Radio Wave Propagation through Ice</b>	<b>10</b>
3.1 Electrodynamics in Vacuum . . . . .	10
3.2 Electrodynamics in Dielectrics . . . . .	11
3.3 Permittivity of Ice . . . . .	18
3.3.1 Physical Properties . . . . .	18
3.3.2 Crystalline Structure of Ice and Conductivity Mechanism . . . . .	18
3.3.3 Dielectric Properties of Monocrystalline and Polycrystalline Ice . . . . .	21
3.4 Ices and Subsurface Salt Water Ocean on Enceladus . . . . .	26
3.4.1 Terrestrial and Nonterrestrial Water Ice . . . . .	26
3.4.2 Stratification of Terrestrial Glaciers and on Enceladus . . . . .	29
3.4.3 Terrestrial Seawater and Subsurface Salt Water Ocean on Enceladus . . . . .	29
<b>4 Methods of Simulating the Radio Wave Propagation through a Dielectric Medium</b>	<b>31</b>
4.1 Motivation and Goals . . . . .	31
4.2 The Parabolic Equation (PE) . . . . .	31
4.2.1 Derivation of the Wide-Angle Parabolic Equation (WAPE) . . . . .	32
4.2.2 Split-Step Parabolic Equation (SSPE) Fourier Transform Solution . . . . .	34
4.2.3 Summary and Interpretation . . . . .	36
4.3 Finite-Difference Time-Domain (FDTD) . . . . .	36
4.4 Ray Tracing (RT) . . . . .	37
4.5 Dipole Antennae . . . . .	38
4.6 Codes . . . . .	39
<b>5 Radio Wave Simulation in a Realistic Scenario on Enceladus</b>	<b>40</b>
5.1 Methods of Analysis . . . . .	40
5.1.1 A-Scan processing: Simulation of the Propagation Time . . . . .	40

---

5.1.2	B-Scan processing: Borehole Ground-Penetrating Radar . . . . .	42
5.1.3	Single-Layer-Scattering . . . . .	43
5.2	Simulation Geometries . . . . .	44
5.2.1	Reference Simulation for Signal Filtering . . . . .	49
5.2.2	Refractive Index of the Simulation . . . . .	50
5.3	Results for the Small-Scale Simulation . . . . .	50
5.4	Results for the Large-Scale Simulation . . . . .	54
5.5	Discussion . . . . .	60
<b>6</b>	<b>Conclusion and Outlook</b>	<b>61</b>
<b>A</b>	<b>Simulation Results</b>	<b>62</b>
<b>B</b>	<b>Conductivity of Enceladean Surface Ice and Subsurface Salt Water Ocean</b>	<b>63</b>
<b>C</b>	<b>Simulation Environment</b>	<b>68</b>
<b>D</b>	<b>Codes</b>	<b>70</b>
<b>E</b>	<b>Bibliography</b>	<b>98</b>
	<b>Acknowledgements</b>	<b>104</b>
	<b>Declaration</b>	<b>105</b>

# List of Figures

1.1	Location of Enceladus . . . . .	1
2.1	Internal structure of Enceladus . . . . .	2
2.2	Enceladus' SPT (a) and false-color image of the SPT (b). . . . .	4
2.3	Sketch of the operating principle of Enceladus' South Polar Plumes . . . . .	5
2.4	Mass Spectrum of Enceladus' plume . . . . .	6
2.5	Logo of the EnEx-initiative . . . . .	7
2.6	A probable scenario of a lander with an ice melting probe onboard . . . . .	7
2.7	Three Phases of the radar-based imaging on Enceladus . . . . .	9
3.1	Dispersive behavior of the complex permittivity . . . . .	22
3.2	Cole-Cole plot of the dielectric function of fresh water at 0 °C. . . . .	25
3.3	Schematic of sintered particles and sinter neck formation of bronze particles . . . . .	28
3.4	Sintering timescales for different temperatures and celestial bodies in the Solar System . . . . .	28
3.5	Stratification of Terrestrial Glaciers and on Enceladus . . . . .	29
3.6	Sketch of a possible explanation for the abundance of methane . . . . .	30
4.1	Simulation of the EM waves in a refractive index profile from Taylor Dome Ice Sheet in Antarctica with random density fluctuations included. . . . .	35
4.2	Staggered lattice (Yee-grid) used in FDTD . . . . .	37
4.3	Advanced incoming ray that is assumed to be straight within each layer . . . . .	38
4.4	Radiation Pattern of half-wave dipole antennae . . . . .	38
5.1	Example of a Gaussian Pulse . . . . .	40
5.2	Gaussian Pulse sampled at 500 MHz for 1000 ns with a pulse center of 200 ns and an amplitude of 1.0 (top) FFT of the the pulse that converts from the time domain into the frequency domain (bottom). . . . .	41
5.3	Schematic representation of a crosshole GPR measurement . . . . .	42
5.4	GPR-based trasillumination surveys . . . . .	43
5.5	Sketch for the derivation of the One-Layer-Scattering formula . . . . .	44
5.6	Positions of RX and TX and 1D Index of Refraction Profile for the pure ice scenario and a flat water pocket . . . . .	45
5.7	Absolute field calculated with the split-step solution of the standard parabolic equation with a source at 25 m depth in the ice, a flat water pocket at 50 m and a central frequency of $f_c = 500$ MHz. . . . .	45
5.8	Simulation Geometry S2: Flat Water Pocket . . . . .	46
5.9	Simulation Geometry S3: Triangular Water Pocket . . . . .	47
5.10	Simulation Geometry S4: Triangular Water Pocket with additional sintered snow layer . . . . .	47
5.11	Simulation Geometry S5: Triangular Water Pocket and Circular Meteoric Rocks . . . . .	48
5.12	B-Scans at a distance of $R = 20$ m between TX and RX in S1. . . . .	49

5.13	Linear Fit of the TOFs across the receiving antenna range in order to determine the refractive index of simulation $n_{simul}$ .	50
5.14	A-Scans of the unfiltered-, reference-, and filtered signal (blue, yellow, red) and derived water pocket depths (blue) of the small-scale simulation in ice.	52
5.15	Crosshole GPR (or Parallel Depth Scans) across the whole range in $x$ -direction.	53
5.16	Fixed Depth Scans. The TX is at a fixed depth of 20 m and the RX is lowered across the whole depth range in $z$ -direction. The colorbar has a wider extent as in the parallel depth scan plots.	54
5.17	Calculated water pocket depths in dependence on the transceiver depth. The red line gives the average of the 25 values.	54
5.18	A-Scans of S2 showing the water reflection spectrum in the time domain at a distance of $R = 495$ m.	56
5.19	Crosshole GPR with ZOP (or Parallel Depth Scan) in the large-scale simulation S3 based on the 4-layer structure with a triangular water pocket on Enceladus.	57
5.20	Absolute Field of S1 (a-c), S3 (d-f) and S5 (g-i) with a single transmitter at different source depths, no receivers and run at a single frequency of $f = 500$ MHz	58
5.21	Absolute field of S3 at a source depth of 200 m	59
5.22	Absolute field of S4 at a source depth of 200 m	59
B.1	Temperature profile model of Enceladus' ice crust	67
C.1	Snow thickness and 2D topography map of the Damascus Sulcus Tiger Stripe	68
C.2	Cumulative plume particle deposition on Enceladus' surface	69

# List of Tables

2.1	Three different types of plume particles . . . . .	6
3.1	Properties of Pure Water Ice ( $f = 500$ MHz, $T = -7^{\circ}\text{C}$ ). . . . .	18
5.1	Labels used for the simulations in this work. . . . .	45
5.2	Complex permittivities and refractive indices calculated via Eq. 3.30 for the simulations S1-5 . . . . .	48
5.3	Transmission ( $t$ ) and reflection ( $r$ ) coefficients for the simulations S1-5 . . . . .	49
A.1	Calculated water pocket depths in a range of 2 to 50 m depth in increments of 2 m. . . . .	62

# Acronyms

- AsGAR** Abbildungssystem für Gletscherspalten in der Astrobiologie zur roboterbasierten Eis-Exploration (engl. Imaging system for crevasses for robot-based ice exploration and astrobiology). 8
- BHR** Borehole Radar. 42
- CDA** Cosmic Dust Analyzer (Cassini instrument). 6, 29, 40
- CIRS** Composite Infrared Spectrometer (Cassini instrument). 2
- DLR** Deutsche Luft- und Raumfahrtgesellschaft. 7, 8
- EnEx** Enceladus Explorer. 7–9
- FDTD** Finite-Difference Time-Domain. 1, 31, 36, 37, 61
- FFT** Fast Fourier Transform. 41
- GPR** Ground Penetrating Radar. iii, vi, 1, 42, 53, 57, 60
- INMS** Ion and Neutral Mass Spectrometer (Cassini instrument). 6, 30
- ISS** Imaging Science Subsystem (Cassini instrument). 3, 4
- PE** Parabolic Equation. 31, 36, 37, 44, 55, 60, 61
- RT** Ray Tracing. 1, 31, 37, 44
- RX** Receiving Antenna. ii, 41, 42, 44, 46, 49, 51, 53, 55, 61
- SAR** Synthetic Aperture Radar. 9, 42
- SPT** South Polar Terrain. ii, 2–4, 46
- TOF** time-of-flight. 9, 40, 50, 51, 55, 61
- TX** Transmitting Antenna. ii, 38, 39, 42, 44, 49, 51, 53, 55, 61
- VIMS** Visual and Infrared Mapping Spectrometer (Cassini instrument). 4, 6
- ZOP** Zero Offset Profiling. iii, 42, 57

## English

Several simulations of radio wave propagation in multiple-layer dielectric media were performed using a Parabolic Equation (PE) method. This was applied to a model of the subsurface environment of the Saturnian moon Enceladus, since radar exploration with an ice melting probe is of huge astrobiological interest due to the possible existence of a subglacial salt water ocean. The dielectric properties of the icy and aqueous layers were derived from Cassini data by considering salinity of the water, impurity level and snow depth of the surface ice layer and possible existing sintering processes. In order to find the depth of a water-bearing crevasse (“water pocket”), a Borehole Ground Penetrating Radar (GPR) method was tested and a simplistic geometric formula was derived that provides a result that deviates 2.84 % from the true value. Reflections from the water were analyzed by so-called A-Scans and B-Scans in the time domain. The simulation can be enhanced by including boundary conditions for irregular and more complex shaped reflection targets and uneven terrains and by adding very-wide-angle approximations as the PE models the energy within a cone in a certain angular range.

## Deutsch

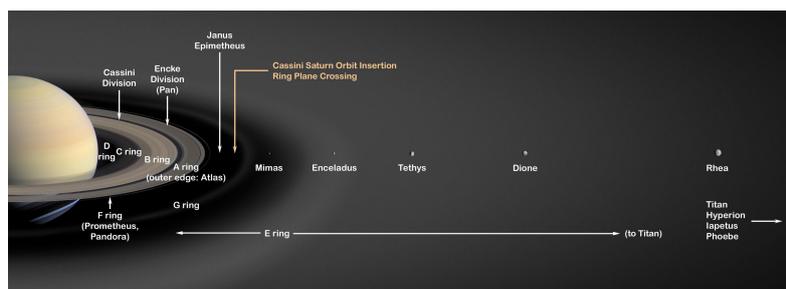
Es wurden mehrere Simulationen der Ausbreitung von Radiowellen in mehrschichtigen dielektrischen Medien mit Hilfe einer Parabolic Equation (PE) Methode durchgeführt. Dies wurde auf ein Modell der unterirdischen Umgebung des Saturnmonds Enceladus angewandt, da die Radaruntersuchung mit einer Schmelzsonde aufgrund der möglichen Existenz eines subglazialen Salzwasserozeans von großem astrobiologischen Interesse ist. Die dielektrischen Eigenschaften der eisigen und wässrigen Schichten wurden aus Cassini-Daten abgeleitet, indem der Salzgehalt des Wassers, der Verunreinigungsgrad und die Schneehöhe der Oberflächeneisschicht sowie mögliche Sinterungsprozesse berücksichtigt wurden. Um die Tiefe einer wasserführenden Gletscherspalte zu bestimmen, wurde eine Methode des Borehole Ground-Penetrating Radar getestet und eine vereinfachte geometrische Formel abgeleitet, die ein Ergebnis liefert, das um 2,84 % vom wahren Wert abweicht. Die Reflexionen des Wassers wurden durch sogenannte A-Scans und B-Scans analysiert. Die Simulation kann durch die Einbeziehung von Randbedingungen für unregelmäßige und komplexer geformte Reflexionsziele und unebenes Terrain sowie durch das Hinzufügen von Näherungen für große Winkel verbessert werden, da die PE die Energie innerhalb eines Kegels in einem bestimmten Winkelbereich modelliert.

## Introduction

The Saturnian moon Enceladus is a prominent candidate for harboring microbial life below the ice sheet in a global salt-water ocean. From an astrobiological perspective, three criteria must be complied to host microbial organisms (McKay et al., 2008):

- existence of organic compounds
- liquid water
- (thermal) energy source

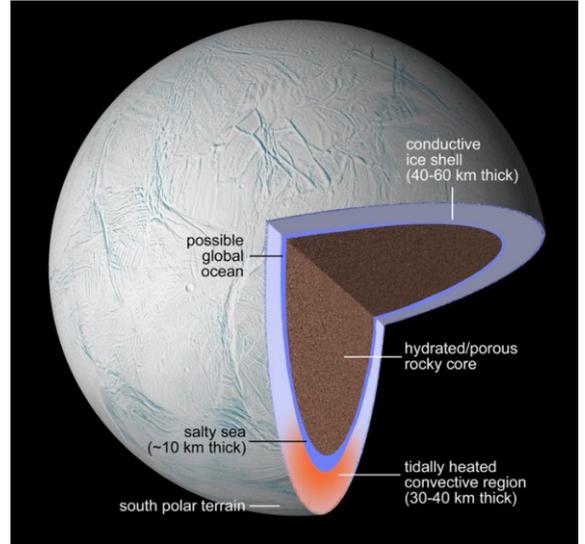
All three criteria could be fulfilled on Enceladus. Organic compounds were detected in the plumes by the Cassini spacecraft's instruments between 2004 and 2017. The existence of plumes, observations regarding the recorded temperature at the south pole of the icy moon and calculations of tidal forces suggest, among other aspects, a global saltwater ocean beneath the surface. Since gases such as hydrogen and methane were also detected in the plumes, this reveals the presence of hydrothermal vents, which might be an energy source in the environment of the deep sea floor and, as seen on Earth, transform the water in an habitat where methanogenic microorganisms are likely to exist (Affholder et al., 2021). The exploration of the interior of Enceladus in terms of chemical and astrobiological aspects has become a major field of research. Several projects are already underway that deal with a possible space mission to the moons of the outer solar system. For a subsurface exploration with a melting probe, there are different approaches (radar, magnetic field measurements, acoustics). This work will deal with simulations of radio wave propagation in dielectric media such as ice using a new method, the parabolic equations, and test its suitability compared to other methods such as Finite-Difference Time-Domain and Ray Tracing. To that end, the dielectric properties for the different types of ices and the saline water on Enceladus will be derived and summed up in a simulation geometry. The distance to a water-bearing crevasse is going to be derived by using simplistic geometry considerations and a radar technique called Borehole GPR.



**Figure 1.1:** Location of Enceladus in the E-Ring of Saturn between the icy moons Mimas (which could possess a subglacial ocean as well) and Tethys. Source: NASA/JPL (2005)

## Enceladus

Enceladus, discovered on 28th of August 1789 by William Herschel, is one of five mid-sized satellites and orbits Saturn between Mimas and Tethys within the E-Ring (cf. Fig. 1.1) and is located about 10 AU from the sun. It has a diameter of 504 kilometers. The icy moon has a bulk density of about  $(1.609 \pm 5) \text{ kg m}^{-3}$  and a rock-to-ice ratio of roughly 60:40 (Spencer and Nimmo, 2013). Enceladus has the highest albedo in the Solar System and can therefore be designated as the brightest satellite. It reflects approximately 80 percent of the sunlight that it intercepts which is related to the deposition of ice-rich plume particles and will be discussed later. Therefore, the reflectance spectrum is dominated by pure water ice. Besides its high albedo, Enceladus is distinguished by its youthful and mainly uncratered surface (Spencer and Nimmo, 2013) including some tectonic rifts and folded ridges (Porco et al., 2006). This is in contrast to most moons of its size in the solar system where the surfaces are covered in impact craters, due to the absence of any geological processes over billions of years. It has the largest range in crater number density of all Saturnian satellites. The region with the lowest crater density is the **South Polar Terrain (SPT)**, which is situated at  $\sim 55^\circ$  latitude in the southern hemisphere of the celestial body and covers an area of  $70.000 \text{ km}^2$ . The most notable feature is that the region is disrupted by Y-shaped discontinuities, the so-called **Tiger Stripes**, which consist of four sub-parallel fissures (named Alexandria, Cairo, Baghdad and Damascus Sulcus). Moreover, each of these depressions, flanked on each side by low ridges, are about 130 km long, 2 km wide, 500 m deep and 35 km spaced apart. More importantly, they reveal hydrothermal activity in the form of **Cryovolcanism** (see below). Another unique characteristic of the SPT is the anomalously high temperature (recorded by CIRS) which reaches local peaks up to 157 Kelvins, especially the ice along the Tiger Stripes, which is significantly warmer than the expected mean surface temperature of Enceladus of 68 Kelvins, if only heat coming from sunlight is considered (Konstantinidis et al., 2015). The elevated heat flux output gives scientifically strong evidence for one or more present heat sources in the interior of Enceladus. According to current research, this is most probably due to tidal heating (Dougherty et al., 2009) which could drive the geological activity (Efroimsky, 2018) and keep a most likely existing subsurface ocean warm for billions of years (Choblet et al., 2017).



**Figure 2.1:** Internal structure of Enceladus. Between an icy crust and a porous rocky core there might exist a global salty ocean with roughly 10 km thickness at the south pole. The SPT is tidally heated up (Porco et al., 2014).

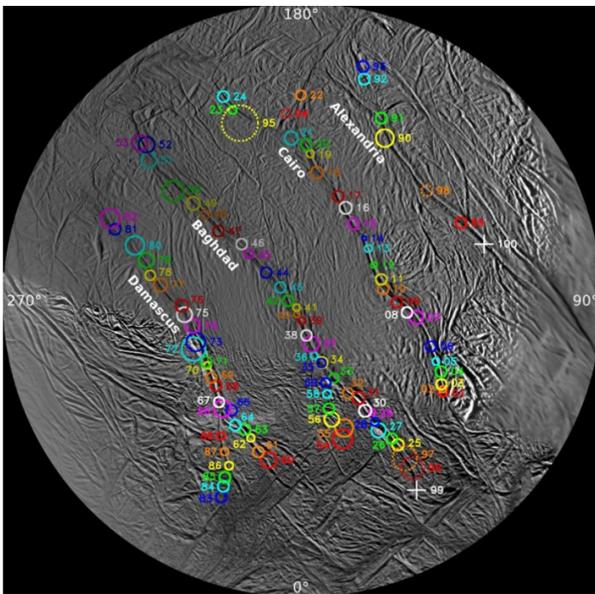
## 2.1 Structure

Enceladus is a differentiated body, which means that it develops compositionally distinct layers due to their geochemical behavior. According to planetary differentiation mechanisms, the layers of different elements then stratify from heavy to light from the inside out: the differentiation of the rock-mass from the (water-) ice-mass demonstrate this, creating a core and a mantle. The reason why the layers were heated and differentiated is because of the abundance of short-lived radionuclides (Castillo et al., 2005) like Al-26 and Fe-60, which generated a lot of heat through radioactive decay. Enceladus, which formed quickly after the formation of the Saturnian subnebula, was rich with these radionuclides (Castillo et al., 2006) as compared to longer-lived radionuclides, and therefore prevented the rapid freezing of the interior and raising the core's temperature up to 1300 K. By means of this heat, both, the core and the ice, partly began to melt and formed magma chambers, accompanied by differentiation of the core. Tidal dissipation from resonance with neighbor satellite Dione and from libration maintained over a long time-scale (Murray and Dermott, 1999) occur in these magma reservoirs, as well as in the ice layer. These friction processes maintain these hot spots stable and drive geological activity, perhaps provided by hydrothermal circulation at the interface between the core and the ice (Matson et al., 2006). Regions beyond the asteroid belt are rich in H<sub>2</sub>O (more precisely: in hydrogen and oxygen). Additionally, Saturn is located beyond the snow line where the environment is cold enough for volatile compounds such as water to condense into solid ice grains. Both of the aforementioned statements explain the occurrence of the thick water ice shell of Enceladus. The most common model of the interior of Enceladus is described as follows, starting from the inside out: the core consists of silicate minerals and is likely irregular shaped, cold, rigid and porous. The radius can derived from bulk density calculations and is estimated at 150-170 km (Spencer and Nimmo, 2013). The core is expected to possess water as a consequence of its porosity. During Cassini's third encounter, approaching Enceladus up to 168 km on 14th of July 2005, scientists observed plumes consisting of H<sub>2</sub>O vapor and salt-bearing ice grains emanated via cryovolcanism from warm fractures at the Tiger Stripes, implying a subsurface liquid salt-water reservoir or even a global ocean below the homogeneous brittle ice crust (Porco et al., 2006). Libration measurements show that the ice crust is fully detached from the rocky core through the water ocean. This ocean has probably a depth of 10 kilometers at the South Pole and lies beneath a water-ice crust (NASA, 2018), whereby the thickness of the crust is less at the poles (5-10 km) than at the equator (30-40 km). In vicinity of the geysers, the terrain is tectonically younger. With increasing distance to them the surface is older and more densely cratered.

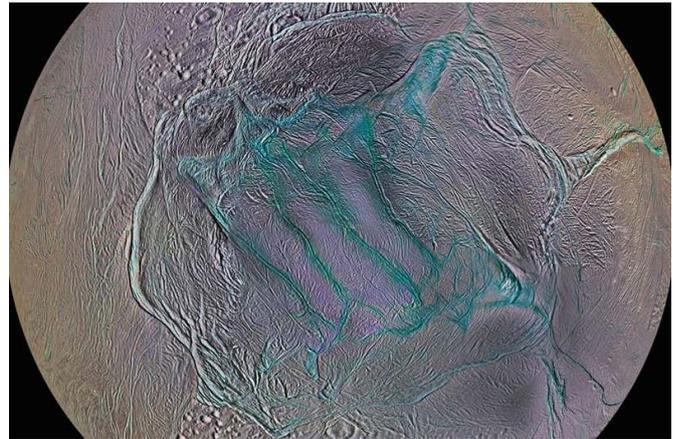
## 2.2 Cryovolcanism and Surface Composition

The Tiger Stripes were first observed on 20th of May 2005 by the ISS camera of Cassini's spacecraft. Over 207 active cryovolcanoes/geysers, that are associated with these four parallel fractures along the SPT, have been identified on Enceladus, including six hotspots (Spitale and Porco, 2007) and 115 weaker sources that were elaborated and proposed by Spitale et al. (2015) and Southworth et al. (2019) by analysing Cassini images with triangulation among other things. The Tiger Stripes appear in a bluish color (cf. Fig. 2.2b) on a false-color image because the ridges in their vicinity are mainly covered with

coarse-grained crystalline water ice which absorbs the near-infrared light used to create the red-channel (Spencer and Nimmo, 2013). On the other hand, nearly the entire surface of Enceladus is coated in a layer of fine-grained pure water ice that in some places reflects 99 percent of the sunlight, renewing the outermost layer again and again before interstellar dust and micrometeorites contaminate the ice. The same applies to impacts of larger meteorites, to which all planets, including Enceladus, are exposed over time, which create larger craters before the depressions are then levelled again by the global renewal of the surface. Globally, the typical water-ice grain size is 50 to 150  $\mu\text{m}$ , but grain sizes increase to 100 to 300  $\mu\text{m}$  in the tiger stripes region (Brown et al., 2006). However, the data situation is not clear: while Degruyter and Manga (2011), Scipioni et al. (2017), Jaumann et al. (2008) assume an average particle size of 75  $\mu\text{m}$  in the Tiger Stripe region, Southworth et al. (2019) uses particle sizes of 0.5-15  $\mu\text{m}$  in order to simulate the snow accumulation rate. In vicinity of the Tiger Stripes, these deposits could also subject to sintering due to the higher heat flux. Further explanations for the varying particle sizes and sintering processes on icy moons will be given below. Additionally, VIMS also detected traces of free carbon dioxide ice, trapped  $\text{CO}_2$  and simple organics (Brown et al., 2006). The geyser-like eruptive activity is in form of many narrow discrete jets (i.e. individual, collimated sources that emerge from the fractures) of micron-sized water ice particles emanating from the SPT, feeding a large plume (i.e. the entire south polar emission composed of all jets and diffuse sources along fractures) and supplying Saturn’s E-ring (Postberg et al., 2018). According to recent knowledge the material sputtered out is not only in form of discrete jets but to a large extent more in form of broad, curtain-like eruptions (Spitale et al., 2015). Based on models of geyser particle deposition on the surface the depth of the unconsolidated snow particles and the sintering level can be estimated (Southworth et al., 2019). The snow depth turns out to be of order 100 m thick (Friend and Kyriacou, 2021).



(a) Polar stereographic basemap of Enceladus’ SPT showing the four Tiger Stripes Damascus, Baghdad, Cairo and Alexandria Sulcus. The numbers indicate the ID numbers of all 100 jets and the colored circles are  $2\sigma$  uncertainties of their location (Porco et al., 2014).

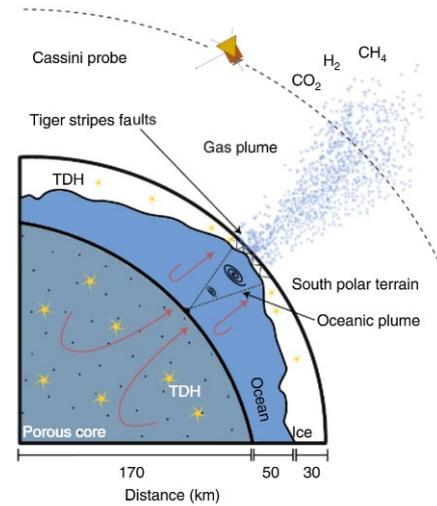


(b) The SPT on a false-color ISS image showing the Tiger Stripes in a bluish color. The region in vicinity of the fractures is mainly coated with coarse-grained crystalline water ice (while at greater distance the surface is coated with fine-grained water ice) which absorbs the near-infrared light used to create the red channel (Sokol, 2017).

**Figure 2.2:** Enceladus’ SPT (a) and false-color image of the SPT (b).

## Enceladus' South Polar Plumes

Generally, a plume on Enceladus is composed with three different phases: Gas, solids (dust) and ions, whereby neutral gas is most abundant and all three phases consisting mainly of water. The gas phase shows a horizontal variation of neutral gas density. This means the plume consists of supersonic collimated high velocity components (“jets”) and much slower outgassing components throughout the whole south polar fractures. More precisely, this means emissions of heavier molecules (with masses  $>200$  u) show a smaller lateral spread than lighter molecules (e.g.  $\text{H}_2$ ,  $\text{H}_2\text{O}$ ), i.e. the spatial composition of high-velocity molecules is mass dependent. If the jet is assumed as a supersonic cone, then the cone angle, equivalent with the terminus of “spreading”, is wider for lower-mass molecules emitted at the same velocity and in thermal equilibrium (Postberg et al., 2018). The ejection speed of water vapor ( $\approx 700$  m/s), which is much greater than the Enceladean escape velocity, is also much higher than that of dust grains, thus, the gas escapes completely into space. The ionic component of the plume interacts with Saturn’s magnetosphere and therefore escapes the gravitation of Enceladus as well. However, from the solid material with grain sizes larger than  $0.1 \mu\text{m}$ , only a small fraction falls back and sediments to the surface, while a greater part escapes to feed the E-Ring. This is because the escape probability is coupled with grain size. Therefore, the deposition rates vary with the surface location, resulting in a vertical stratification in particle size: the largest particles were found in vicinity of the geyser fractures while smaller particles have been found further away. Note, that larger particles have lower speeds and therefore reach lower travel heights and distances away from the plume whereas smaller particles have higher speeds and land further away in a greater distance to the geysers, and therefore are more likely to sediment most of Enceladus’ surface. The smallest particles have the highest speeds and escape into space to feed the E-Ring. The larger near-surface particles are generally more enriched in salts and are possibly frozen droplets of salty liquid water, while smaller particles are more poor in salts and therefore consist mostly pure water-ice that generated by the condensation of relatively salt-free plume vapor. However, there still exist an uncertainty for both, the origin of those molecules sprayed out of the cryovolcanoes and the mechanism of the geyser-like activity.

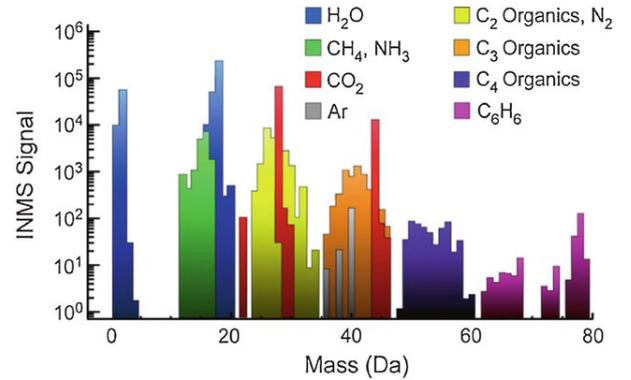


**Figure 2.3:** Tidal dissipative heating (TDH) in the core (yellow stars) and water circulation in the core and in the ocean (red arrows). The plumes are shaped by buoyancy and Coriolis forces (black spirals). The Cassini spacecraft is represented in one of its fly-bys of Enceladus, during which it crossed the gas plume of oceanic material escaping into space (dotted trajectory) (Affholder et al., 2021).

## Chemical Composition of Enceladus' Plumes

The majority of the ice grains are crystalline and not amorphous which implies a formation at temperatures above 130 Kelvins and emphasizes the temperature anomaly. The determination of the chemical, elemental and isotopic composition of the gaseous and volatile components of the neutral particles and the low energy ions in Saturn's magnetosphere and in the ring environment could have been analyzed by INMS. Direct measurements of particles of any kind encountering the spacecraft could have been taken by CDA, while with VIMS spectra constraining particle size and composition and including surface deposits around the plume sources could have been generated. In advance, the lack of sensitivity of the Cassini instruments for heavier and more complex molecules must be considered (Postberg et al., 2018).

Besides 98 % of water ( $\text{H}_2\text{O}$ ), the most frequently occurring constituents and volatiles in the vapor are represented by a mixture of ammonia ( $\text{NH}_3$ , 0.4-1.3 %), molecular hydrogen ( $\text{H}_2$ , 0.4-1.4 %), carbon dioxide ( $\text{CO}_2$ , 0.3-0.8 %) and organics, whereas the most abundant organics in the gas component is methane ( $\text{CH}_4$ , 0.1-0.3 %) (Waite et al., 2017) apart from  $\text{C}_2$ , Ar,  $\text{C}_3$  and  $\text{C}_4$ . There might as well be O- and N-bearing organics in both, neutral gas and ice grains. Directly measured molecules in ionized form are represented mainly by water and water products, i.e. cations in form of  $\text{H}_n\text{O}^+$  ( $n=0-3$ ), anions in the form of  $\text{OH}^-$ ,  $\text{H}^-$  or  $\text{O}^-$  and water clusters in form of  $(\text{H}_2\text{O})_n\text{OH}^-$  ( $n=1-3$ ). The abundance of high-mass anions ( $m > 200$  u) consistent with complex organics, N-bearing cations (e.g.  $\text{NH}^+$ ) and cations with masses of 28 u (maybe  $\text{N}_2^+$  and/or  $\text{CO}^+$ , cf. Fig. 2.4) is not yet conclusively clarified but are in well agreement with an ample scientific evidence of measurements in Saturn's magnetosphere. About 99 % of the mass in the solid components of the jets is salt-rich (Porco et al., 2014). The plume particles are geochemically similar to the components of comets (Battersby, 2008), which basically incorporate a conglomerate of rock, dust, water ice and light organics, but differ in their proportions. The most abundant neutral, non-icy compounds in the solid phase for plume particles with radii greater than  $0.2 \mu\text{m}$  are sodium salts and organics. The sodium-salts-bearing particles, mainly consisting of water ice and sodium chloride ( $\text{NaCl}$ ) ( $0.05-2 \text{ mol kg}^{-1}$ ) and sodium hydrogen carbonate ( $\text{NaHCO}_3$ , 2-5 times less than  $\text{NaCl}$ ), can be subdivided into three types (cf. Tab. 2.1):



**Figure 2.4:** Mass spectrum of the Enceladus plume from the October 9th 2008 flyby. The colors show contributions from various species (Waite Jr et al., 2009)

**Table 2.1:** Three types of ice grains that were detected and identified differently often, which differ in weight, size and chemical composition (Postberg et al., 2009)

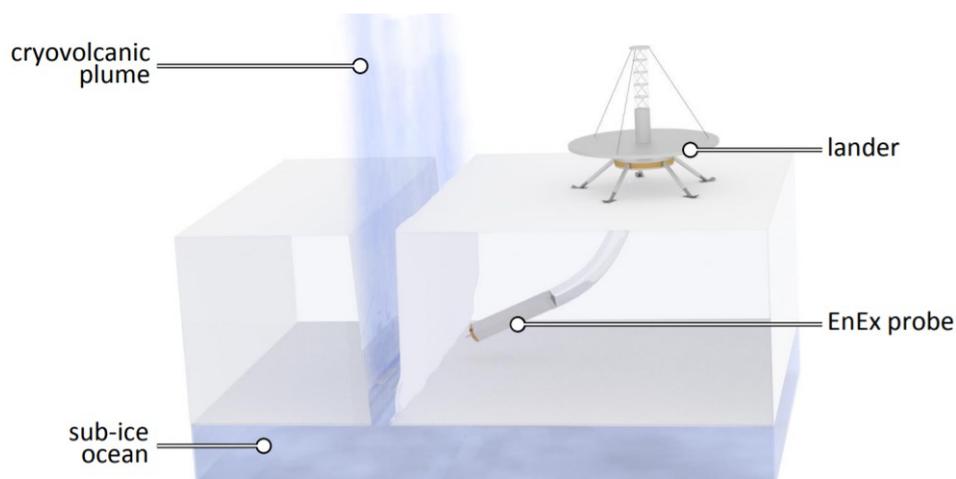
	Type I	Type II	Type III
Weight/Size	light/small	light/larger than Type I	heavy/large
Composition	Pure Water Ice	similar to Type I + organics	Traces of N + K salts
$\frac{\text{Na}}{\text{H}_2\text{O}}$	$10^{-8} - 10^{-5}$	similar to Type I	$> 10^{-3}$ , molality 0.5-2%

## 2.3 EnEx - Enceladus Explorer

Enceladus, which is within reach for mankind, could harbor extraterrestrial life. In order to explain the origin of the organic compounds detected by Cassini, it would be useful to examine the water directly in the interior while it is still liquid. Traces of microbial life that might have developed in a possible salt-water ocean would burst and likely not survive the (freezing) process while rising through the ice fissures due to pressure, especially not when ejected from the geysers and exposed to the hostile conditions on the Enceladean surface. From 2012 to 2015, the Space Administration of the German Aerospace Center (DLR-Raumfahrtmanagement) conducted the Enceladus Explorer (EnEx) project. Within this project, in total of 8 distinct institutes and universities worked as partners on the development of key technologies for the exploration of Enceladus and possibly other icy moons in the outer solar system, one of which was to develop a robotized ice melting probe (Konstantinidis et al., 2015). One of the most the challenging



**Figure 2.5:** Logo of the EnEx-initiative. Source: DLR



**Figure 2.6:** A lander with an ice melting probe onboard that lands near a geyser-active fracture. The probe melts and drills through the ice until it reaches the inner wall of an up-going channel, takes samples of the liquid plume material and do microbiological analysis (Eliseev, 2018).

aspects is the navigability and especially the autonomy of the probe that has to be realized, since the signal transit time from Earth to Enceladus is about two hours, depending on the positions of Earth and Saturn. The EnEx melting probe will basically be based on the concept of the IceMole, which was developed by the FH Aachen and whose functionality has successfully be proven in a field test in Antarctica, where researchers extracted an uncontaminated water-sample from the Blood Falls in 2015. After the completion of the Verbundvorhaben, the EnEx-initiative was launched, whereby the DLR coordinates a couple of parallel running, individual sub-projects that are strongly linked to each other. Each of them focuses on a specific technology aspect with the main goal to develop systems for the navigation of a maneuverable subsurface ice melting probe for clean sampling and in-situ analysis of ice and subglacial liquids. They use different approaches for the navigation like radar, acoustics or magnet field measurements.

## Mission to Enceladus and Requirements for the Melting Probe

A possible mission to Enceladus could look like this: after launch, a lander and an orbiter perform the interplanetary transfer together carrying the melting probe. Once arrived close enough to Enceladus, the orbiter would decouple from the lander, which then in turn would land at a safe distance away of an active plume. The orbiter would be used for the purpose of communication between lander and Earth. The melting probe would then melt and drill through the ice and navigate towards a subglacial aquiferous fracture autonomously while circumnavigating hazards and obstacles. After taking samples at a depth of 100-200 meters, the probe, which is directly connected to the base station via cable for power supply, would send this information to the surface. Summing up the abovementioned, the following requirements and enhancements can be placed on the melt probe (Konstantinidis et al., 2015): It should

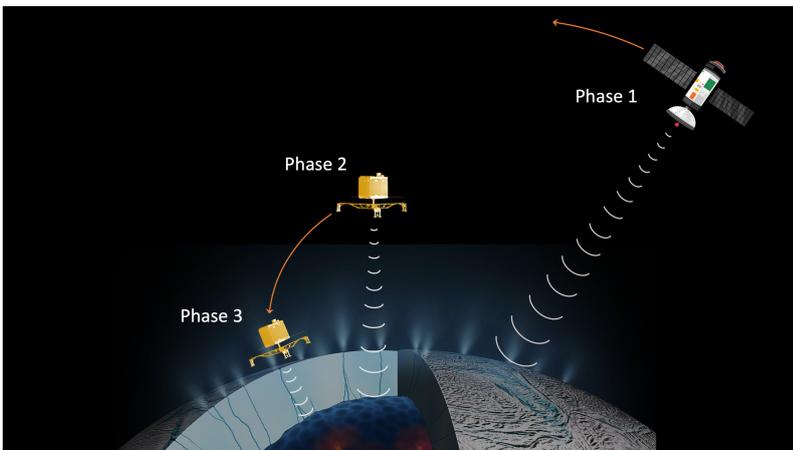
- possess a melting, drilling and propulsion mechanism so that the probe can steer in any direction,
- be able to locate a liquid water-bearing crevasse over a distance of  $>100$  m,
- be autonomous and self-learning: It should be able to plan and follow an optimal path from the surface to the calculated destination completely independently,
- sense and avoid obstacles in the ice like cavities and rocks which could hinder further movement,
- continuously determine its actual position in the ice relative to the surface station and to the target. It should also take into account range and energy expenditure,
- sample liquid water, do in-situ analysis in terms of testing the water on biosignatures and send the data to the surface station.

However, the work on the mission concept is still in progress and design features are not final but could change until a final version of the concept is defined.

### EnEx-AsGAR

One of the sub-projects within EnEx focuses on radar-based mapping of the ice, especially developing an imaging system for crevasses for robotic ice exploration to ensure a suitable landing site. It is called EnEx-AsGAR and is a collaboration of University of Wuppertal (BUW), Friedrich-Alexander University Erlangen-Nuremberg (FAU) with the support from the DLR-HR in Oberpfaffenhofen. The aim is to generate high resolution and spatially accurate images of the subsurface ice environment in front of the melting probe during and after landing for which the ice parameters (see below) are needed. Accordingly, the radar-based exploration on Enceladus is divided into three phases (cf. Fig. 2.7) and would look like the following:

- **Phase 1: Remote Sensing.** Coarse resolution detection and mapping of the surface and subglacial structure of geyser cracks from the orbit to determine an appropriate landing position and gauging the ice depth.
- **Phase 2: Descent.** Improved structure resolution and imaging of deeper structures during descent of the lander to the surface in order to locate for example a water-bearing crevasse.
- **Phase 3: Near-field imaging.** Fine resolution recording of the subsurface ice. The EnEx melting probe melts and penetrates through the ice, navigates towards its target and takes in-situ measurements while the radar supports its localization by SAR principle. Identification of obstacles like meteoric rocks.



**Figure 2.7:** Three Phases of the radar-based imaging on Enceladus. Phase 1 and 2 is for the purpose of long-range imaging in order to determine a suitable landing spot while in Phase 3 the short-range imaging would give more details of the subsurface in terms of obstacles and the water pocket(s). Credit: Dr. Pia Friend

Commonly used techniques of radar systems can not be applied for the mission since the EnEx-lander is limited in mobility and a line-by-line scanning of the ice region with highly directional radar arrays are not possible. Hence, a novel imaging concept has been developed which is principally based on synthetic aperture radar (SAR), whose kernel is expanded with Fermat's principle of least time. This has already been exercised in remote sensing tests where glaciers land areas of several kilometers have been captured by an aircraft's overflight with help of SAR technique. SAR can be applied by making

the assumption of a ground that appears flat from great height or when using the perpendicular radar system with strong directivity. In this case (Phase 1), the refractions of the electromagnetic waves at the ice-vacuum boundary layer, occurring because of the different dielectric constants of vacuum and ice, can be neglected. This is different during and after landing. The surface becomes uneven and the examination area small, so that refraction and reflection effects on the ice surface and in the subsurface ice have to be considered that have to be compensated by the radar system with help of an improved imaging algorithm of SAR. By means of this reflection at the boundary layer a 3D map of the surface structure of Enceladus can be generated. These radar mapping techniques require the knowledge of the permittivity  $\epsilon_r$ , since propagation speed through the ice and refraction angle at the surface boundary (scattering) and energy absorption (attenuation) are dependent on it. The permittivity is determined by parameters such as salinity, density (porosity) and temperature of the ice, which are not entirely known for Enceladus' ice crust. Radar localization is based on the fact that the radar velocity is well-known in air or vacuum. Both, the propagation speed and the time-of-flight (TOF) of incident waves, allow to calculate the distance to obstacles like meteoric rocks and water-bearing crevasses.

## Radioglaciology - Radio Wave Propagation through Ice

In this chapter, properties of dielectric material will be discussed. It will focus specifically on radio wave propagation in ice. The following conventions will be used:

$$\begin{aligned}\vec{\nabla} \cdot \vec{E} &\equiv \text{div } \vec{E} - \text{divergence of } \vec{E} & \dot{\vec{E}} &\equiv \frac{\partial E}{\partial t} - \text{time derivation of } \vec{E} \\ \vec{\nabla} \times \vec{E} &\equiv \text{rot } \vec{E} - \text{rotation of } \vec{E} & \vec{\nabla} \cdot \vec{\nabla} &= \Delta - \text{Laplace operator}\end{aligned}$$

$\vec{E}$  is the electric field vector,  $\rho$  is the charge density,  $\vec{B}$  is the magnetic field vector,  $\vec{j}$  is the current density vector,  $\epsilon_0 = 8.85 \cdot 10^{-12} \text{ A} \cdot \text{S} \cdot \text{V}^{-1} \cdot \text{m}^{-1}$  is the permittivity or dielectric constant in free space and  $\mu_0 = 4\pi \cdot 10^{-7} \text{ N} \cdot \text{A}^{-2}$  is the magnetic permeability of free space.

### 3.1 Electrodynamics in Vacuum

The core of Electrodynamics is formed by the microscopic MAXWELL's Equations. They are fundamentally important for describing the dynamics of the propagation of electromagnetic waves in vacuum.

#### MAXWELL'S EQUATIONS (microscopic)

$$\vec{\nabla} \cdot \vec{E} = \frac{\rho}{\epsilon_0} \quad \vec{\nabla} \times \vec{E} = -\dot{\vec{B}} \quad (3.1)$$

$$\vec{\nabla} \cdot \vec{B} = 0 \quad \vec{\nabla} \times \vec{B} = \mu_0 \vec{j} + \mu_0 \epsilon_0 \dot{\vec{E}} \quad (3.2)$$

Let the charge- and current density vanish so that the electric and magnetic fields are free of sources. Then, Ampères circuital law with Maxwell's addition (Eq. 3.2) can be written as:

$$\dot{\vec{E}} = \frac{1}{\mu_0 \epsilon_0} \vec{\nabla} \times \vec{B} \quad (3.3)$$

Now take the derivation of Equation 3.3 and insert the result in Maxwell-Faraday equation (Eq. 3.1). Finally, it is:

$$\begin{aligned}
 \ddot{\vec{E}} &= \frac{1}{\mu_0 \varepsilon_0} \vec{\nabla} \times \dot{\vec{B}} \\
 &= -\frac{1}{\mu_0 \varepsilon_0} \vec{\nabla} \times (\nabla \times \vec{E}) \\
 &= -\frac{1}{\mu_0 \varepsilon_0} (\vec{\nabla}(\vec{\nabla} \cdot \vec{E}) - (\vec{\nabla} \cdot \vec{\nabla})\vec{E}) \\
 &= \frac{1}{\mu_0 \varepsilon_0} \Delta \vec{E}
 \end{aligned}$$

where the vector triple product identity  $\vec{\nabla} \times (\vec{\nabla} \times \vec{E}) = \vec{\nabla}(\vec{\nabla} \cdot \vec{E}) - (\vec{\nabla} \cdot \vec{\nabla})\vec{E}$  is used and the electric field is assumed to be free of divergence:  $\vec{\nabla} \cdot \vec{E} = 0$ . Let us write this equation in a more explicit form:

$$(\mu_0 \varepsilon_0 \partial_t \partial_t - \partial_i \partial_i) \vec{E} = 0 \Leftrightarrow \square \vec{E} = 0 \quad (3.4)$$

where  $\square$  is the d'Alembert operator. This equation is called **Wave equation**. The negative sign transforms the elliptic Laplace Equation in a new hyperbolic equation which allows different solutions. The ansatz

$$\vec{E}(z, t) = \vec{E}(z \pm ct) \quad (3.5)$$

is a solution of 3.4 if the constant is set to  $c = \frac{1}{\sqrt{\mu_0 \varepsilon_0}}$ . The electric field remains unchanged in his form but propagates at a velocity of

$$c = \frac{1}{\sqrt{\mu_0 \varepsilon_0}} \simeq 2.998 \cdot 10^8 \frac{\text{m}}{\text{s}} \quad (3.6)$$

which is identified as velocity of light in free space (vacuum) (Hoelbling, 2020, pp. 73-75). An explicit form of the solution for the wave equation can be  $\vec{E}(z, t) = E_{0,z} \cdot e^{-ikz}$ . Hence, the time-harmonic wave is a solution to the wave equation derived from Maxwell's equation since  $k = \frac{\omega}{c}$ , where  $\omega = 2\pi f$  is the circular frequency and controls the time evolution of the wave, while  $k = \frac{2\pi}{\lambda}$  controls the spatial evolution. Consequently, it is

$$k_0 = \frac{2\pi f}{c} = 2\pi f \sqrt{\varepsilon_0 \mu_0} \quad (3.7)$$

## 3.2 Electrodynamics in Dielectrics

So far, electrostatic potentials and fields in the presence of charges and conductors were discussed where there was no difference between microscopic and macroscopic fields. Thus, in vacuum or air it was allowed to neglect their dielectric properties because they are sufficiently thin as a medium. This is not the case for denser media, where responses to external electric fields have to be considered. It is impossible to describe microscopic electric fields and charge distribution in a dielectric since on average about  $10^{23}$  (sub-)atomic and molecular particles per cubic centimeter are in motion, resulting in spatially and temporarily rapidly oscillating fields. Therefore, it is consulted only the field strength averaged over many molecule diameters, i.e. over macroscopic little but microscopic large areas containing a multitude of atoms/molecules. After averaging, the equation  $\vec{\nabla} \cdot \vec{E}_{\text{micro}} = 0$  maintains its homogeneous shape:

$\vec{\nabla} \cdot \vec{E} = 0$ , where  $\vec{E}$  represents the averaged, macroscopic electric field which is still derivable from a potential. Matter mainly consists of charged particles (protons, electrons, ions,...) which are not always free to move but respond to external fields, which throw them off the equilibrium (Nolting, 2013). If these electric charges are understood as neutral units, such as molecules or a constituents of a crystal lattice, they will not have a total charge but a multipole moment. When a field is applied, the bonded charges in each molecule react to this perturbation and make additional movements, meaning that the molecular charge density gets distorted, respectively, the multipole moment of each molecule changes its value. This multipole moment can be induced by a permanent asymmetric electron cloud, where the positive and negative charges shift relative to each other (displacement polarization) or by an external electric field, where permanent dipoles align (orientation polarization). The induced additional electric field superimposes respectively counteracts the external field and weakens it inside. This property is called dielectricity and materials that possess this property are called dielectrics (Hoelbling, 2020). In absence of an external electric field, the multipole moments vanish in the most substances after averaging. In presence of an external electric field the dipole term is the most dominant term in the multipole expansion. This leads to an *electric Polarization P*:

$$\vec{P}(\vec{x}) = \sum_i N_i \cdot \langle \vec{p}_i \rangle \quad (3.8)$$

where  $p_i$  is the dipole moment of the i-th molecule and  $N_i$  is the average number of molecules of the i-th type per volume unit around  $\vec{x}$ , whereas the averaging is performed around a little volume around  $\vec{x}$ . If the molecules have a total charge  $e_i$  and an excess of free charges the charge density can be written as:

$$\rho(\vec{x}) = \sum_i N_i \langle e_i \rangle + \rho_{\text{free}} = \rho_{\text{bound}} + \rho_{\text{free}} \equiv \rho_b + \rho_f \quad (3.9)$$

Generally, the average molecular charge is equal to zero, hence, the charge density is represented solely by free charges. On macroscopic observation of the medium, the charge density changes to  $\rho \mapsto \rho_f - \rho_P = \rho_f - \vec{\nabla} \cdot \vec{P}$  (effective charge density). The new Maxwell's equation is therefore given by:

$$\vec{\nabla} \cdot \vec{E} = \frac{1}{\varepsilon_0} [\rho - \vec{\nabla} \cdot \vec{P}] \quad (3.10)$$

Let  $\vec{D} = \varepsilon \vec{E} + \vec{P}$  be the electric displacement field, then the counterparts to Maxwell's Equations in free space (Eq. 3.1) is given by:

$$\vec{\nabla} \cdot \vec{D} = \rho \quad \vec{\nabla} \times \vec{E} = -\dot{\vec{B}} \quad (3.11)$$

In order to determine a relation between the fields  $\vec{D}$  and  $\vec{E}$ , the following assumption must be made: the system responds linearly to the external field and the medium is isotropic. Then, the polarization  $\vec{P}$  is parallel to  $\vec{E}$ :

$$\vec{P} = \varepsilon_0 \chi_e \vec{E} \quad (3.12)$$

where  $\chi_e$  is the electric susceptibility and is independent of direction. In this case, the displacement field is proportional to the electric field:

$$\vec{D} = \varepsilon \vec{E} \quad (3.13)$$

with  $\varepsilon = \varepsilon_0(1 + \chi_e)$ . For homogeneous media,  $\varepsilon$  is location-independent and the left side of Equation 3.11 can be written as:

$$\vec{\nabla} \cdot \vec{E} = \frac{\rho}{\varepsilon} \quad (3.14)$$

which means that a macroscopic problem can be reduced to a microscopic problem if the electric field is reduced by the factor of  $\frac{\varepsilon}{\varepsilon_0}$  because of the counteracting inner electric fields. For inhomogeneous material,  $\varepsilon$  and  $\mu$  depend on location and perhaps time and for anisotropic materials (e.g. crystal structures) these scalars become tensors ( $\varepsilon_{ik}, \mu_{ik}, \chi_{ik}$ ). Analogous considerations can also be made for magnetostatics. The current density as a function of location was well-known so far. Since a multitude of electrons move in the atoms of matter, this is not the case anymore, as a consequence of effective atomic currents, whose density fluctuates. Besides, atoms possess inner magnetic moments which generate dipole fields that vary over a significant atomic area. Like already shown, the averaging over such an area brings us to the macroscopic equations. Again, the divergence of  $\vec{B}$  is equal to zero before as well as after the averaging ( $\vec{\nabla} \cdot \vec{B}_{\text{micro}} = 0$  and  $\vec{\nabla} \cdot \vec{B} = 0$ ) so that a vector potential provides the magnetic field. If the average (macroscopic) magnetization is set to  $\vec{M}(\vec{x}) = \sum_i N_i \langle \vec{m}_i \rangle$  with  $\vec{m}_i$  the average molecular magnetic moment of the  $i$ -th molecule per volume unit around  $\vec{x}$ , then an additional term to the current density is included in the form of  $\vec{j} \mapsto \vec{j}_f + \vec{j}_M$  ( $j_f$  is the macroscopic current density caused by movement of free charges and  $\vec{j}_M = \vec{\nabla} \times \vec{M}$ ). Analogous to above, the current density in Maxwell's equation in free space (Eq. 3.2) is replaced by the new effective current density and it is:

$$\vec{\nabla} \times \vec{B} = \mu_0 [\vec{j}_f + \vec{\nabla} \times \vec{M}] \quad (3.15)$$

Let  $\vec{H} = \frac{1}{\mu_0} [\vec{B} - \vec{M}]$  the magnetic field. The counterpart to the magnetic Maxwell's equations is now given by:

$$\vec{\nabla} \times \vec{H} = \vec{j}_f \quad \vec{\nabla} \cdot \vec{B} = 0 \quad (3.16)$$

For isotropic substances it can be established the following relation between the two fields:

$$\vec{B} = \mu \cdot \vec{H} \quad (3.17)$$

whereby  $\mu$  is the magnetic permeability. The new system of equations is not consistent. For Ampère's circuital law 3.16 it was made the assumption of a stationary system, i.e.  $\vec{\nabla} \cdot \vec{j} = 0$  but has to be replaced with a continuity equation for time dependent fields:  $\vec{\nabla} \cdot \vec{j} + \dot{\rho} = 0$ . Add the displacement current and get  $\vec{j} \mapsto \vec{j}_f + \dot{\vec{D}}$ . The final and correct Maxwell's equations in a dielectric medium are given by:

MAXWELL'S EQUATIONS (macroscopic)

$$\vec{\nabla} \cdot \vec{D} = \rho_f \quad \vec{\nabla} \times \vec{E} = -\mu \dot{\vec{H}} \quad (3.18)$$

$$\vec{\nabla} \cdot \vec{B} = 0 \quad \vec{\nabla} \times \vec{H} = \vec{j}_f + \dot{\vec{D}} \quad (3.19)$$

whereby Ampère's circuital law 3.19 can be also written as  $\vec{\nabla} \times \vec{H} = \sigma \vec{E} + \varepsilon \dot{\vec{E}}$  where  $\vec{j} = \sigma \vec{E}$  and  $\sigma$  is the conductivity. The essential point here is that a varying electric field generates a magnetic field even when no current is flowing, which was important to understand the physics of light and electromagnetic waves. In accordance to Equation 3.7, for a wave travelling through any material other than a vacuum, and taking into consideration that the phase velocity can be defined as  $v_\phi = \frac{\omega}{k}$ , it is:

$$k = \frac{2\pi f}{v_\phi} = 2\pi f \sqrt{\varepsilon \mu} \quad (3.20)$$

Hence, in accordance with the velocity in vacuum (Eq. 3.6) the phase velocity is now given by

$$v_\phi = \frac{1}{\sqrt{\mu \varepsilon}} = \frac{1}{\sqrt{\varepsilon_0 \varepsilon_r \mu_0 \mu_r}} \quad (3.21)$$

where

- $\mu_0 = 1.26 \cdot 10^{-6}$  H/m - absolute magnetic permeability
- $\varepsilon_0 = 8.85 \cdot 10^{-12}$  F/m - absolute electric permittivity of free space
- $\mu = \mu_0 \mu_r$  - absolute magnetic permeability of medium
- $\varepsilon = \varepsilon_0 \varepsilon_r$  - absolute electric permittivity of medium
- $\varepsilon_r$  - relative permittivity ( $\approx 1-80$  for most geologic materials)
- $\mu_r$  = relative magnetic permeability (1 for nonmagnetic geologic materials)

Another important quantity is the refractive index, which describes how fast light travels through the material and is defined as

$$n = \frac{c}{v} = \sqrt{\mu_r \varepsilon_r} \quad (3.22)$$

respectively, if  $\mu_r \approx 1$ , so that magnetic interactions are neglected, then

$$n \approx \sqrt{\varepsilon_r} \quad (3.23)$$

Hence the phase velocity of a wave propagating in a dielectric medium is:

$$v_r = \frac{c}{\sqrt{\varepsilon_r}} \quad (3.24)$$

The electromagnetic waves consist of an electric and a magnetic component. In a perfect dielectric the magnetic susceptibility and the electric permittivity are constants, which means that they are independent from frequency and the medium is not dispersive. Additionally, there are no attenuation losses. This is not so for real dielectrics. However, in material of interest of surface-penetrating radar the magnetic losses are negligible (i.e. it is not a complex quantity) whereas the electric losses has to be considered,

which means the permittivity  $\varepsilon$  and conductivity  $\sigma$  are complex quantities:

$$\varepsilon = \varepsilon' - i\varepsilon'' \quad (3.25a)$$

$$\sigma = \sigma' - i\sigma'' \quad (3.25b)$$

The real part  $\varepsilon'$  is linked to the electric permittivity while the imaginary part  $\varepsilon''$  is associated with losses in conductivity and frequency due to absorption. Therefore, Equation 3.23 can be rewritten as:

$$n \approx \sqrt{\varepsilon' + i\varepsilon''} \quad (3.26)$$

For oscillating electrical fields that induce a frequency dependence of the polarization it is required to change the dielectric constant to the (complex) dielectric function that now contains all necessary information:

$$\varepsilon_r(\omega) = \varepsilon'(\omega) + i\varepsilon''(\omega) \quad (3.27)$$

Consequently, it makes sense to do so for the refractive index as well. Instead of using  $n'$  and  $n''$  for the real and imaginary part, which is historically justified,  $n$  and the extinction coefficient  $\kappa$  will be used:

$$n^*(\omega) = n(\omega) + i\kappa(\omega) \quad (3.28)$$

Using the relation given in Equation 3.23 this yields to:

$$(n + i\kappa)^2 = \varepsilon' - i\varepsilon'' \quad (3.29)$$

Bringing the real and imaginary parts of the complex permittivity and the complex refractive index together, one can write them in dependence of each other:

$$n^2 = \frac{1}{2} \left( (\varepsilon'^2 + \varepsilon''^2)^{\frac{1}{2}} + \varepsilon' \right) \quad \kappa^2 = \frac{1}{2} \left( (\varepsilon'^2 + \varepsilon''^2)^{\frac{1}{2}} - \varepsilon' \right) \quad (3.30)$$

and

$$\varepsilon' = n^2 - \kappa^2 \quad \varepsilon'' = 2n\kappa \quad (3.31)$$

The electric loss tangent is defined as

$$\tan \delta_e = \frac{\varepsilon''}{\varepsilon'} \quad (3.32)$$

and the loss tangent fulfils the equation

$$\tan \delta = \frac{\omega\varepsilon'' + \sigma}{\omega\varepsilon' - \sigma''} \quad (3.33)$$

The new wave equations induced by the macroscopic Maxwell's equations are given by:

$$\Delta \vec{E} = \mu\sigma \dot{\vec{E}} + \mu\varepsilon \ddot{\vec{E}} \quad (3.34)$$

$$\Delta \vec{H} = \mu\sigma \dot{\vec{H}} + \mu\varepsilon \ddot{\vec{H}} \quad (3.35)$$

For sinusoidal time-varying fields (time-harmonic fields), Equations 3.34 and 3.35 can be written as

$$\Delta \vec{E} = \gamma^2 \vec{E} \quad (3.36)$$

$$\Delta \vec{H} = \gamma^2 \vec{H} \quad (3.37)$$

and is called HELMHOLTZ Wave Equation where the complex number

$$\gamma = \sqrt{i\omega\mu(\sigma + i\omega\varepsilon)} = \alpha + i\beta \quad (3.38)$$

is called propagation constant and includes the attenuation constant  $\alpha$ , which is related to the decrease of the amplitude and the phase constant  $\beta$  that characterizes the propagation of the wave. Both quantities can be expressed in terms of the permeability, the conductivity and the permittivity:

$$\alpha(\omega) = \omega \sqrt{\frac{\mu\varepsilon}{2} \left[ \sqrt{1 + \left(\frac{\sigma}{\omega\varepsilon}\right)^2} - 1 \right]} \quad \beta(\omega) = \omega \sqrt{\frac{\mu\varepsilon}{2} \left[ \sqrt{1 + \left(\frac{\sigma}{\omega\varepsilon}\right)^2} + 1 \right]} \quad (3.39)$$

The phase constant  $\beta$  can be used in order to determine the phase velocity analogous to  $k$ :

$$v_\phi = \frac{\omega}{\beta} = \frac{1}{\sqrt{\frac{\mu\varepsilon}{2} \left[ \sqrt{1 + \left(\frac{\sigma}{\omega\varepsilon}\right)^2} + 1 \right]}} \quad (3.40)$$

From here, two limit value observations can be performed: insulators ( $\sigma \rightarrow 0$ ) and conductors ( $\sigma \rightarrow \infty$ ):

$$\lim_{\sigma \rightarrow 0} \alpha(\omega) = \frac{\sigma}{2} \sqrt{\frac{\mu}{\varepsilon}} \rightarrow 0 \quad (3.41)$$

$$\lim_{\sigma \rightarrow 0} \beta(\omega) = \omega \sqrt{\mu\varepsilon} = \frac{\omega}{v_\phi} = k \quad (3.42)$$

$$\lim_{\sigma \rightarrow 0} v_\phi(\omega) = \frac{1}{\sqrt{\mu\varepsilon}} \quad (3.43)$$

since  $\lim_{\sigma \rightarrow 0} \sqrt{1 + \left(\frac{\sigma}{\omega\varepsilon}\right)^2} = 1$ . These general forms reduce exactly to the originally derived equations for free space. There is no dissipation ( $\alpha = 0$ ) and  $\beta$  reduces to the wave number  $k$ .

$$\lim_{\sigma \rightarrow \infty} \alpha(\omega) = \sqrt{\frac{\omega\mu\sigma}{2}} \rightarrow \infty \quad (3.44)$$

$$\lim_{\sigma \rightarrow \infty} \beta(\omega) = \sqrt{\frac{\omega\mu\sigma}{2}} \rightarrow \infty \quad (3.45)$$

$$\lim_{\sigma \rightarrow \infty} v_\phi(\omega) = \frac{2\omega}{\mu\sigma} \rightarrow 0 \quad (3.46)$$

The general form of the solution to Equation 3.36 for a wave polarized in the y-direction and originating at  $z = 0$ ,  $t = 0$  is:

$$E(z, t) = E_{0,z} \cdot e^{-\gamma z} = E_{0,z} \cdot e^{-(\alpha+i\beta)z} = E_{0,z} \cdot e^{-\alpha z} \cdot e^{-i\beta z} = E_{0,z} \cdot \underbrace{e^{-\alpha z}}_{\text{attenuation term}} \cdot \underbrace{e^{i(\beta z - \omega t)}}_{\text{propagation term}} \quad (3.47)$$

The propagation of a plane electromagnetic wave in positive z-direction can also be described by

$$E(z, t) = \Re \left[ E_{0,z} \cdot e^{-i(kz - \omega t)} \right] \quad (3.48)$$

where  $k_z$ , as introduced before, is the component of the wave vector in z-direction. It describes the change in phase per unit length for each wave component. and fills the role of  $\beta$  in Equation 3.47. In order to put the electric field in dependence of the refractive index, the known expression for the propagation velocity can be used. It is  $v = \frac{c}{n^*}$  and  $k = \frac{2\pi}{\lambda} = \frac{\omega \cdot n^*}{c}$ . Insertion yields

$$\begin{aligned} E(z, t) &= \Re \left[ E_{0,z} \cdot \exp \left\{ i \cdot \left( \frac{\omega \cdot n^*}{c} \cdot z - \omega \cdot t \right) \right\} \right] \\ &= \Re \left[ E_{0,z} \cdot \exp \left\{ i \cdot \left( \frac{\omega \cdot (n + i \cdot k)}{c} \cdot z - \omega t \right) \right\} \right] \\ &= \Re \left[ E_{0,z} \cdot \exp \left\{ \left( \frac{i \cdot \omega \cdot n}{c} - \frac{\omega \cdot \kappa \cdot z}{c} - i \cdot \omega \cdot t \right) \right\} \right] \\ E(z, t) &= \underbrace{E_{0,z} \cdot \exp \left\{ -\frac{\omega \cdot \kappa \cdot z}{c} \right\}}_{\text{Decreasing amplitude}} \cdot \underbrace{\Re \left[ \exp \left\{ i \cdot (k_z \cdot z - \omega \cdot t) \right\} \right]}_{\text{Plane wave}} \end{aligned}$$

in compliance with Equation 3.47. In concrete terms this means with use of a complex refractive index the propagation of electromagnetic waves in a material is whatever it would be for an ideal material with only a real index of refraction *times* an attenuation factor that decreases the amplitude exponentially as a function of depth z. If  $z = \frac{c}{\omega} \cdot \kappa := W$ , the intensity decreases by a factor of  $\frac{1}{e}$  and is known as absorption length or penetration depth. The imaginary part, on the other hand, is known as damping constant, attenuation index, extinction coefficient or (rather misleading) absorption constant. If  $\eta = \frac{E}{H}$  is defined as the intrinsic impedance, then it is possible to describe received signals that scattered in multi-layer dielectrics at corresponding boundaries. At these boundaries, some energy is reflected and some is transmitted. This can be described by the reflection ( $r$ ) and transmission ( $t = 1 - r$ ) coefficients. If a continuous transition of two dielectric media 1 and 2 with impedances  $\eta_{1,2}$  and with relative permittivities  $\varepsilon_{r1,2}$  is assumed, then (Daniels, 1996, pp. 34-42):

$$r = \frac{\eta_2 - \eta_1}{\eta_2 + \eta_1} = \frac{\sqrt{\varepsilon_{r2}} - \sqrt{\varepsilon_{r1}}}{\sqrt{\varepsilon_{r2}} + \sqrt{\varepsilon_{r1}}} \quad t = \frac{2\eta_1}{\eta_2 + \eta_1} = \frac{2\sqrt{\varepsilon_{r1}}}{\sqrt{\varepsilon_{r2}} + \sqrt{\varepsilon_{r1}}} \quad (3.49)$$

The transmission and reflection of light of a wave incident on an interface between two different optical media at an arbitrary angle can be described by the FRESNEL equations:

$$r = \left| \frac{n_2^* \cos \alpha - n_1^* \cos \beta}{n_2^* \cos \alpha + n_1^* \cos \beta} \right|^2 \quad t = \left| \frac{2n_1^* \cos \alpha}{n_2^* \cos \alpha + n_1^* \cos \beta} \right|^2 \quad (3.50)$$

In case for a nonconducting medium and when considering only a single frequency of radiation and an incident angle of  $0^\circ$ , where there is no distinction between the types of polarization, the formulas can be changed to:

$$r = \frac{(n_2 - n_1)^2 + (\kappa_2 - \kappa_1)^2}{(n_2 + n_1)^2 + (\kappa_2 + \kappa_1)^2} \quad t = 1 + \frac{(n_2 - n_1)^2 + (\kappa_2 - \kappa_1)^2}{(n_2 + n_1)^2 + (\kappa_2 + \kappa_1)^2} \quad (3.51)$$

### 3.3 Permittivity of Ice

#### 3.3.1 Physical Properties

Generally, ice is water frozen into solid state. The properties of ice<sup>1</sup> (cf. Table 3.1) can vary substantially with temperature, impurity, porosity and other factors.

**Table 3.1:** Properties of Pure Water Ice ( $f = 500$  MHz,  $T = -7^\circ\text{C}$ ).

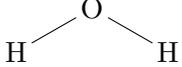
Density ( $\rho$ )	Refractive index ( $n^*$ )		Dielectric constant ( $\epsilon_r$ )	
	$n$	$\kappa$	$\epsilon'$	$\epsilon''$
0.91617 $\frac{\text{g}}{\text{cm}^3}$	1.7861	0.0002	3.1902	0.0007

Ice is a naturally occurring solid substance, has a well-defined crystallographic structure and has a fairly well-defined chemical composition, thus, according to International Mineralogical Association, it is considered to be a mineral (Nickel and Grice, 1998). There exist currently 19 types of solid crystalline phases of water that are known which distinguish in their density and crystal system.

#### Types of Ice

- \* Amorphous ice - ice without crystalline structure, i.e. without long-range order
- \* Ice I<sub>h</sub> - hexagonal crystalline ice
- \* Ice I<sub>c</sub> - cubic crystalline ice where the oxygen atoms are arranged in a diamond structure (formation temperatures of 130 K up to 220 K)

#### 3.3.2 Crystalline Structure of Ice and Conductivity Mechanism

The water molecule (H<sub>2</sub>O) is composed of two hydrogen atoms and one oxygen atom:  Its shape is in form of an isosceles triangle, since an asymmetric order would not be stable. The molecule can form, because oxygen has six valence electrons and therefore shares two pairs of electrons with hydrogen in a water molecule. These are called homeopolar or covalent bonds. They result in a positively charged net around in vicinity of the hydrogen atoms that distorts the electron clouds around the oxygen atom and therefore leads to a concentration of negative charge on the opposite side. Consequently, the positions of the two hydrogens correspond to two of four vertices of a tetrahedron centered on a oxygen atom whereas the other two vertices correspond to the centers of the negative electron clouds. One has four centres of electric charge (2+, 2-) corresponding to four vertices of a tetrahedron altogether (Glen and Paren, 1975). Notwithstanding the existent electron clouds surrounding the hydrogen, they are not strong enough to neutralize the positive charge. In total, water is a neutral molecule with a large dipole moment induced by the aforementioned orientation polarization. The legs of the triangle are not

<sup>1</sup><https://refractiveindex.info/?shelf=3d&book=crystals&page=ice>, Last access on 26th of June 2021

arranged rectangular because of the repulsion between the positive charges of the two hydrogen atoms. As a consequence of this repulsion, the legs of the triangle are forced open beyond  $104.5^\circ$  at a distance of 95.8 pm between H and O. Because of its polarity, the water molecules can form up hydrogen bonds between the negative charge concentration and positive charge concentration of one molecule to another below the boiling point. They are about 10 times stronger than Van der Waals forces that attract atoms or molecules to each others. In liquid phase, the hydrogen bonds form up and break simultaneously but with decreasing thermal motion by cooling, these bonds pull the molecules together to a rigid crystalline structure of ice whereas the apex increase to  $109.5^\circ$ . Each oxygen with its for nearest neighbours build a four-fold tetragonal order which leads to sheets of crinkled hexagons stacked one on top of another Further, the density decreases, respectively, the volume occupied by the molecules increases because the packing of molecules less compact in the solid (Bogorodsky et al., 2012).

### Bernal-Fowler rules

Under normal pressure and temperature conditions, the nuclei have no crystallographic arrangement, meaning that there exist no long-range order but are randomly arranged except for some requirements. In 1933 the British physicists John D. BERNAL and Ralph H. FOWLER in essence proposed two rules that should describe the arrangement of atoms in water ice, since the molecular vibration of ice was found to be similar to those of a free water molecule in Infrared and Raman spectra (Bogorodsky et al., 2012):

1. There is one and only one proton per O-O linkage (binding force between water molecules)
2. There are two and only two protons associated with each oxygen atom (ice is not ionized)

with the exception of a proton disorder by high residual entropy of the ice. The electrical properties of ice are not defined by the crystalline structure per se, but by the violation of that structure. The breaches of the Bernal-Fowler rules, namely, producing point defects in the structure of ice which first was described by Niels BJERRUM (Glen, 1968). Three kinds of crystallographic defects specific to ice were postulated:

1. Violation of tetrahedral bonding (occurrence of vacancies or interstitial molecules)
2. Orientational defect: Violation of the first BF rule. Occurrence of either
  - i two protons (D-Defect, *from Danish: Dobbelt = double*) or
  - ii none proton (L-Defect, *from Danish: Ledig = empty*)
 along the O-O bond.
3. Ionic defect: Violation of the second BF rule. Occurrence of either
  - i one proton (resulting in  $\text{OH}^-$ ) or
  - ii three protons (resulting in  $\text{H}_3\text{O}^+$ )

associated with individual oxygen atom.

The ionic defects arise from proton jumps from one end of an O-O bond to the other, resulting in an ionized state that heaves the energy level and is metastable. More importantly, the finite time to jump from one bond to the other by overcoming the energy barrier, in small fraction of cases, allows the proton to remain in the displaced position long enough for another proton from the same molecule to jump to a third oxygen. Thus, the oxonium ion and the hydroxide ion, produced in pair, are separated by a simple H<sub>2</sub>O which allows them to move free through the crystal until they encounter ions charged with the opposite sign and get destroyed. Bringing all this together, the electrical conduction in ice depends upon the movement of these ion states through the crystal. The orientational defects, which are produced in pair too, represent the more important defects (1 per 10<sup>7</sup> molecules versus 1 per 10<sup>12</sup> molecules). A proton jumps and “surpasses” the other proton of the water molecule leaving a hole at the O-O bond, which now is occupied by no proton, while the O-O bond of the other molecule linkage is now occupied by two protons. Alternatively, one can describe this process by a rotation of a molecule around one of its bond through an angle of 120°. Again, the finite time of this jump or rotation facilitates the stabilization of the defect pair through another proton jump (or rotation). Both, the process of ionic and the orientational defects are thermally activated, therefore their concentration will be proportional to a Arrhenius function  $\exp\{-\frac{\Delta H}{k_B T}\}$ , where  $\Delta H$  is the activation enthalpy (energy). Its generation requires 0.7 eV and its migration through the crystal 0.25 eV at a temperature of -10°C.

#### Defects due to Impurities

Defects can arise from impurities. In this case the water molecule is doped with foreign atoms. For example, let us take hydrogen fluoride (HF). On the one hand, one can substitute HF with H<sub>2</sub>O as a whole. Then, one bond which had one of two hydrogen of H<sub>2</sub>O has no more hydrogen, which complies with a L-defect with no corresponding D-defect and is the dominating defect in ice at temperatures near the melting point. On the other hand, consider the chemical reaction of  $\text{HF} + \text{H}_2\text{O} = \text{F}^- + \text{H}_3\text{O}^+$ , which obviously corresponds to an ionic defect (Glen and Paren, 1975).

#### Summary

Jaccard’s theory of conduction in an ice crystal sums up, as already discussed above. When an external electric field is absent, equal numbers of tetrahedra will be oriented in all orientations allowed by the ice structure so that there is no net electric dipole moment. But if an electric field is applied, then it is energetically favourable for dipoles to be aligned in a particular direction whose number is given by statistical mechanics of the system (Barnes et al., 2002). Conduction takes place by protons moving (diffusing) through the ice lattice under influence of the electric field and enabled by the presence of lattice defects. They produce point defects and rotating/reorienting water molecules as they pass that partly gives ice its dielectrical properties such as permittivity. The concentration of these defects is temperature-dependent but can also be increased by doping the ice crystal with impurities, generating the ionic and Bjerrum L- and D-defects. The speed of response to the electric field is correlated to the number and mobility of those L defects, which then determine the relaxation time (Jaccard, 1964).

## Nonwater Ices

Ice on Enceladus is widely pure water ice over the entire surface, with the exception of the SPT. There, a non-vanishing fraction of CO<sub>2</sub> in the ice was detected. It is scientifically not proven yet, if it occurs as a pure phase, is enclathrated with water ice (i.e. incorporated in the ice crystal) or adsorbed on other species, or a combination of all three. From measurements, the following electromagnetic properties of CO<sub>2</sub> ice are known in the MHz-GHz range (Pettinelli et al., 2015):

- $\rho = 1.50 \text{ g cm}^{-3}$
- $\varepsilon = 2.12 \pm 0.05$
- $\sigma = 10^{-9} - 10^{-7} \text{ S m}^{-1}$

### 3.3.3 Dielectric Properties of Monocrystalline and Polycrystalline Ice

If an ice crystal is now exposed to an external electric field, one will have two kinds of responses, namely, polarization and conduction. The former takes place by dipoles lining up parallel to the electric field which is caused by the proton jumps that induce orientational defects. Thereby, the bound charges undergo a local redistribution to a new equilibrium position, storing and dissipating energy. The development of this polarization has a time delay, since the protons have to overcome the energy barrier thermally. Therefore, the length of the time delay is dependent on temperature. The latter effect, the conduction, takes place when free charges are present, dissipating energy as well. So far, only stationary electric field were regarded. Let the electric field be time-varying:

$$E = E_0 \cdot e^{i\omega t} \quad (3.52)$$

then, with given permittivity and loss tangent the electrical displacement  $D = \varepsilon E$  can be written as:

$$D = D_0 \cdot e^{i(\omega t - \delta)} \quad (3.53)$$

where  $D_0 = |\varepsilon|E_0$ . The aforementioned polarization and conduction effects can be describe by a sum of density currents  $J_T = J_c + J_d = \sigma_s E + \frac{\partial D}{\partial t}$ , where  $D = \varepsilon_0 + P = \varepsilon_0 \varepsilon E$ ,  $J_c$  the conduction current density and  $J_d$  the displacement current density. If an alternating field with frequency  $\omega$  is applied to the material, the total current density can be written as:

$$J_T = \sigma_s E + i\omega \varepsilon_0 \varepsilon E = \underbrace{(\sigma_s + \omega \varepsilon_0 \varepsilon'')}_{:=\sigma} E + i\omega \varepsilon_0 \varepsilon' E = \sigma E + i\omega \varepsilon_0 \varepsilon' E \quad (3.54)$$

where  $\sigma$  is the effective conductivity which describes the energy loss in the material. Here,  $\sigma_s$  is associated with the conduction of steady (dc) current, thus, the inelastic scattering of free charge carriers during their migration in the material at all frequencies (Pettinelli et al., 2015), wherein the diffusion of either type of defect (orientational or ionic) leads quickly to a fully polarized state, in which proton transfer is impossible. Accordingly,  $\sigma_s$  is the ratio of current density to the applied electric field (Bogorodsky

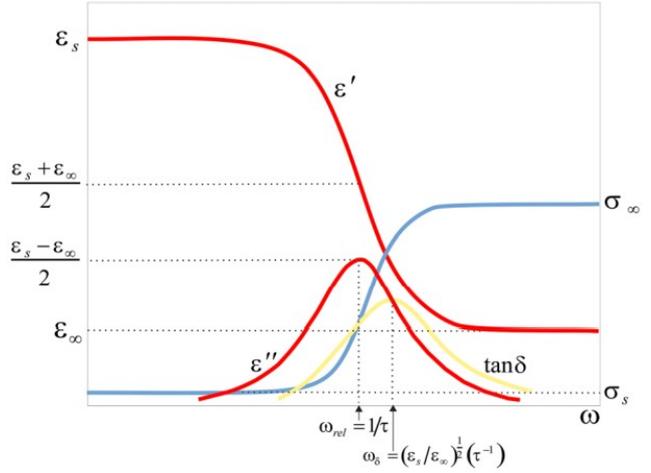
et al., 2012). Further,  $\omega\varepsilon_0\varepsilon''$  is the term describing the friction during the polarization process and increases linearly with  $\omega$  but vanishes for  $\omega = 0$ . The imaginary part  $\omega\varepsilon_0\varepsilon'E$  describes the polarization in the material due to the displacement current. With this knowledge about the static conductivity, the complex permittivity can be redefined:

$$\varepsilon = \varepsilon' - i\left(\varepsilon'' + \frac{\sigma}{\omega\varepsilon_0}\right) = \varepsilon' - i\frac{\sigma}{\omega\varepsilon_0} \quad (3.55)$$

and the loss tangent as well:

$$\tan \delta = \frac{\varepsilon'' + \frac{\sigma_s}{\omega\varepsilon_0}}{\varepsilon'} \quad (3.56)$$

The polarization is proportional to  $\exp\{-\frac{t}{\tau}\}$ , thus, the rate of a proton jumping to a certain orientation and the relaxation time  $\tau$  is dependent on the probability of a protonic jump, described by an Arrhenius function that shows the temperature dependency. Hence, the permittivity shows a dispersive behavior in the frequency domain. The most common model was developed by P. DEBYE and is valid for lots of geomaterials in a broad frequency range up to Gigahertz's. The equations belonging to this model are called standard Debye equations:



**Figure 3.1:** Dispersive behavior of the complex permittivity. (Petinelli et al., 2015)

$$\varepsilon(\omega) = \varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + i\frac{\omega}{\omega_{rel}}} - i\frac{\varepsilon_s}{\omega\varepsilon_0} \quad (3.57)$$

The real and imaginary part of the complex permittivity are now given by:

$$\varepsilon' = \varepsilon_\infty + \frac{\Delta\varepsilon}{1 + (\frac{\omega}{\omega_{rel}})^2} \quad \varepsilon'' = \frac{\sigma_s}{\omega\varepsilon_0} + \frac{\Delta\varepsilon \cdot \frac{\omega}{\omega_{rel}}}{(1 + (\frac{\omega}{\omega_{rel}})^2)} \quad (3.58)$$

where

- $\varepsilon_s, \sigma_s$  - static values of  $\varepsilon'$  and  $\sigma$
- $\varepsilon_\infty, \sigma_\infty$  - high-frequency limit of  $\varepsilon'$  and  $\sigma$ , i.e. alternating current (ac) conductivity
- $\Delta\varepsilon = \varepsilon_s - \varepsilon_\infty$  - dispersion strength
- $\omega_{rel} = \frac{1}{\tau_{rel}}$  - relaxation angular frequency where  $\tau_{rel}$  is the relaxation time ( $\sim$  pico-seconds)

If the amplitude of the real and imaginary parts are plotted against frequency, one will get the so-called idealized Debye spectrum, which indicates the dispersive behavior of the dielectric function. First thing

that can be noted is that  $\varepsilon''$  attains a maximum of  $\frac{\varepsilon_s - \varepsilon_\infty}{2}$  for  $\omega_{\text{rel}} = \tau^{-1}$ , the characteristic relaxation angular frequency (cf. Fig. 3.1). The effective conductivity can be written as  $\sigma = \sigma_s + (\sigma_\infty - \sigma_s) \cdot \frac{\omega^2}{\omega^2 + \omega_{\text{rel}}^2}$ , where  $\sigma_\infty = \sigma_s + \varepsilon_0(\varepsilon_s - \varepsilon_\infty)\omega_{\text{rel}}$  is the high-frequency limit of  $\sigma$ . For  $\omega \gg \omega_{\text{rel}}$  the effective conductivity approaches  $\sigma_\infty$ , the real part of permittivity approach the frequency-independent value  $\varepsilon_\infty$  while the imaginary part behaves like  $\frac{\sigma_\infty}{\omega\varepsilon_0}$  and thus  $\varepsilon'' \rightarrow 0$  for  $\omega \rightarrow \infty$ . **Polycrystalline ice** is a crystalline solid consisting of many crystallites (or grains), i.e. little crystalline sub-structures that are separated by grain boundaries. Their electrical parameters are similar to those of monocrystalline ice and at the anisotropies such as grain boundaries, their values are close to those of the values taken with the electric field perpendicular to the c-axis. However, there are two aspects in which they differ to the monocrystals: the deformational history (such as plastic deformation) and metamorphic history (was the ice formed by direct freezing from liquid state, or by cold metamorphism of snow, or by metamorphism of snow but with subsequent temperatures close to the melting point?). But, the statements made here are only valid for frequencies less than  $\omega_{\text{rel}}$ ; there is no experimental evidence that this is also valid for radar frequencies (Bogorodsky et al., 2012).

### Looyenga Model for Dielectric Correlation between Powder and Bulk

The admittance  $Y$ , which usually is measure for the current flow and the reciprocal of the impedance  $Z$ , can be used to describe the polarization or susceptibility of an ice lattice under the influence of an alternating field. Since it is a complex quantity, it can be written in terms of the conductance  $G(\omega)$ , which corresponds to the real part of the complex conductivity  $\sigma'(\omega)$  and capacitance  $C(\omega)$ , which corresponds to the real part of the relative permittivity  $\varepsilon'(\omega)$ :

$$Y(\omega) = G(\omega) + i\omega C(\omega) \quad (3.59)$$

In the high-frequency limit  $\sigma'(\omega)$  reaches a plateau that is denoted with  $\sigma_\infty$ . Each of the dielectric components can be represented by such an equation of the complex admittance  $y_j$ :

$$y_j = \sigma_j DC + i\omega\varepsilon_0\varepsilon_j \quad (3.60)$$

Models of describing the pure component of the conductivity of polar ice with varying density in depth were adequately discussed in the past. Hence, statements can be made about the dielectric properties of different intermediate states of ice (snow, firn, ...). This is achieved by considering a heterogeneous dielectric mixture of air and ice, where spherical inclusions that form one in the other component are included and cause a change in density. Dependent on this density, i.e. if one consider high-density ice or low-density firn, one may view these spherical inclusions either as air bubbles or as spheres of ice in the air, respectively. First a model of Böttcher was the most widely used (Böttcher, 1952) until both Landau and Lifshitz (1960) and Looyenga (1965) independently proposed a model for a the dielectric correlation between powder and bulk at microwave and radio frequencies, that is, a scenario with such a dielectric mixture dependent on the volume packing fraction  $v$  in order to describe its high-frequency behavior  $\sigma_{\infty \text{ pure}}$ . Let  $\varepsilon$  be the relative permittivity of the mixture and  $\varepsilon_1, \varepsilon_2$  the relative permittivities

of the two components in the mixture. Then models can be described using the following relations:

$$\varepsilon_1^{\frac{1}{3}} - \varepsilon_2^{\frac{1}{3}} = v(\varepsilon_2^{\frac{1}{3}} - \varepsilon_1^{\frac{1}{3}}) \quad [\text{Landau-Lifshitz-Looyenga}] \quad (3.61)$$

$$\frac{\varepsilon - \varepsilon_1}{3\varepsilon} = \frac{v(\varepsilon_2 - \varepsilon_1)}{\varepsilon_2 + 2\varepsilon} \quad [\text{Böttcher}] \quad (3.62)$$

Both of them are suitable in order to describe the permittivity of low-density firn through to solid ice but, for the sake of simplicity, the following will be limited to the Looyenga model. It provides an accuracy of 3 – 8 % for particle sizes less than 30  $\mu\text{m}$  (Dube, 1970). Since a part of the mixture is air,  $\varepsilon_1$  can be set to  $\simeq 1$  and the other other part consists of the pure ice component, thus  $\varepsilon_2 = \varepsilon_{\infty \text{ pure}} = 3.17$  for  $\rho = 920 \text{ kg m}^{-3}$  and  $T = -18^\circ \text{ C}$ . For a low loss tangent above the Debye dispersion frequency of ice, the permittivity of snow can be expressed using the Looyenga equation 3.62:

$$\varepsilon_{\infty, \text{ snow}} = (1 + 0.496v)^3 \quad (3.63)$$

and the high-frequency conductivity of this mixture,  $\sigma_{\infty \text{ Looyenga}}(v)$ , by taking the imaginary part of Equation 3.63 (Glen and Paren, 1975):

$$\sigma_{\infty, \text{ snow}} \simeq \sigma_{\infty, \text{ Looyenga}} = \sigma_{\infty, \text{ pure}}(1) \cdot (0.68 + 0.32v)^2 \quad (3.64)$$

### Conductivity of Ice Crystals containing Impurities

Doping an ice crystal with impurities generally leads to an increase in conductivity. Wolff et al. (1997) identified a linear relationship between the high-frequency limit  $\sigma_{\infty}$  and the acid concentration.  $\sigma_{\infty}$  can then be expressed as a sum of all impurity contributions to the pure ice component ( $\sigma_{\infty, \text{ snow}} \equiv \sigma_{\infty, \text{ pure}}$ ):

$$\sigma_{\infty}(v) = \sigma_{\infty \text{ pure}}(v) + \sum_i \beta_i(v)c_i \quad (3.65)$$

where  $\beta_i$  is the molar conductivity of each impurity component and  $c_i$  is their molar concentration. Now consider firn containing impurities under the influence of a DC current ( $\omega = 0$ ). The admittance (Eq. 3.60) then has no imaginary component. The DC conductivity of air is zero. Subsequently,

$$\sigma_{DC}(v) = \sigma_{DC}(1)v^3 \quad (3.66)$$

The DC conductivity of pure ice is approximately zero. This implies that  $\sigma_{DC}$  corresponds to a net bulk conductivity of both ice crystals and the salt molecules incorporated in or around them.  $\sigma_{DC}(1)$  stands for the non-porous (solid) ice and is dependent on the salt concentration. Overall, this means that  $\sigma_{DC}$  is the DC conductivity if the ice was solid, corrected for density. This model can be applied to the impurity content of the ice (Moore et al., 1989):

$$\beta_i(v) = \beta_i(1)v^3 \quad (3.67)$$

and combined with the correction of Looyenga's model of conduction (Shabtaie and Bentley, 1994):

$$\beta_i(v) = \beta_i(1) \cdot v \cdot (0.68 + 0.32v)^2 \quad (3.68)$$

## Conductivity of Sea Water

The permittivity of saline water is frequency-, temperature- and salinity-dependent. Sea water mainly is fresh water with several dissolved salts and therefore is a good conductor. Thereby the salinity is a parameter that is a measure for the concentration of the ionic salts, that is, the total weight of the dissolved salts in the solution in grams per kilograms or parts per thousands (ppt), respectively. Consequently, the conductivity of water increases approximately to the same extent as ions are added ( $\sigma \propto N_{\text{Ions added}}$ ). A good approximation that is very close to reality for frequencies in the range of 1 to 1000 GHz is given by the Debye model. The augmentation of polarization due to the displacement of bound charges depends on the salinity of the water due to the ions added. Thus,  $\varepsilon_s$  and  $\varepsilon_\infty$  are functions of the sea water's salinity. This will be further elaborated below.

## Debye's Theory

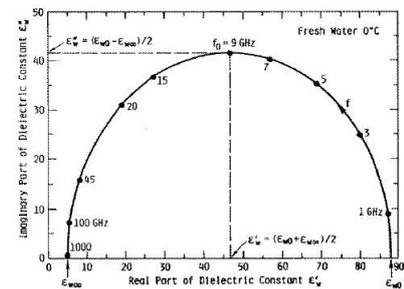
Debye describes the relationship between the relative permittivity  $\varepsilon_r$  and the frequency of electromagnetic waves in a dielectric medium. He makes the assumption that the molecules are free and do not interact to each other. Placing the dielectric substance in an external static electric field leads to polarization (dipole moment per volume). Debye investigated the two contribution of polarizations (i.e. induced and orientation polarization) in terms of the relative permittivity. The induced polarization happens almost instantaneously and takes the value of  $\vec{P}_I = \varepsilon_0(\varepsilon_{\infty-1})\vec{E}$ . The alignment of the molecules instead take a finite amount of time which rises exponentially. The time-constant  $\tau$  of the exponential term is the so-called relaxation time. For  $t \rightarrow \infty$  the orientation polarization attains the value of  $\vec{P}_O = \varepsilon_0(\varepsilon_s - 1)\vec{E} - \varepsilon_0(\varepsilon_\infty - 1)\vec{E}$ . In total, the polarization for a static field is  $\vec{P} = \vec{P}_O + \vec{P}_I = \varepsilon_0(\varepsilon_s - 1)$ . The frequency dependence of  $\varepsilon_r$  is given by the following formula:

$$\varepsilon_r(\omega) = \varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + i\omega\tau} \quad (3.69)$$

but note that the permittivities are functions of temperature, thus:  $\varepsilon_\infty = \varepsilon_\infty(T)$  and  $\varepsilon_s = \varepsilon_s(T)$ . This model is based on the idealisation that there is no intermolecular interaction (see above) so that this formula is highly limited in accuracy. If several relaxation times instead of a single one are allowed (something like a distribution of relaxation times) then it may be better to use Cole-Cole model of permittivity (Cole and Cole, 1941). It is rather an empirical model than a physical one and provides an higher accuracy for plenty of dielectrics:

$$\varepsilon(\omega, T) = \varepsilon_\infty(T) + \frac{\varepsilon_s(T) - \varepsilon_\infty(T)}{1 + (i\omega\tau)^{1-\alpha}} = \begin{cases} \text{Debye-Model, for } \alpha = 0 \\ \text{Model with stretched relaxation, for } \alpha > 0 \end{cases} \quad (3.70)$$

where  $1 - \alpha$  is the Cole-Cole exponent with  $0 \leq \alpha \leq 1$ . In some cases the models by Cole-Davidson,



**Figure 3.2:** Cole-Cole plot of the dielectric function of fresh water at 0 °C. (Source: wikipedia.org (a))

or Havriliak-Negami can fit better to experimental data. Since the water contains impurities, the permittivity, as mentioned above, depends on a third parameter, the salinity. The main goal is to find an interpolation function for  $\sigma(T, S)$  in order to calculate  $\varepsilon_r(\omega, T, S)$  in an aqueous saline solution. Subsequently, the role of the ionic and organic content in sea water regarding the permittivity can be explained. This can be achieved by deducing polynomial interpolation functions for the Debye parameters  $\tau(T, S)$ ,  $\varepsilon_s(T, S)$ ,  $\varepsilon_\infty(T, S)$  through least-squared-regression from experimental data. These parameters are linear functions of S. The average difference between the interpolation and the measurements deviate by less than 1 % and fits perfectly to the permittivity model. The relative permittivity  $\varepsilon_r$  typically can be stated by the following expression:

$$\varepsilon(\omega) = \varepsilon_0 \cdot \varepsilon_r(\omega) = \varepsilon' - i \cdot \varepsilon'' = \varepsilon_0 \cdot (1 + \chi_e) = \varepsilon' \cdot (1 - i \cdot \tan \delta_e) \quad (3.71)$$

and by using the Debye model (cf. Eq. 3.69) with interpolated Debye parameters, separating the real and imaginary part and adding the conductivity term, it can be approximated by (Ellison et al., 1998):

$$\varepsilon_r(\omega, T, S) = \left( \varepsilon_\infty(T) + \frac{\varepsilon_s(T, S) - \varepsilon_\infty(T)}{1 + (\omega \cdot \tau(T, S))^2} \right) - i \cdot \left( \frac{\sigma(T, S)}{\omega \cdot \varepsilon_0} + \frac{\omega \cdot \tau \cdot (\varepsilon_s(T, S) - \varepsilon_\infty(T))}{1 + (\omega \cdot \tau(T, S))^2} \right) \quad (3.72)$$

where  $\omega = 2\pi f$  is the angular frequency,  $\tau$  is the relaxation time (delay of particles to respond to the field change),  $\varepsilon_\infty$  is the high-frequency limit of the permittivity whereas  $\varepsilon_s$  is the low-frequency limit (static permittivity) and  $\varepsilon_\infty$  is a parameter that is well-known for fresh water. It is usually set to  $\approx 81$  at  $T = 25^\circ\text{C}$  (and  $\approx 88$  at  $T = 0^\circ\text{C}$ , cf. Fig. 3.2), while  $\varepsilon_s \approx 4$  (Karagianni, 2015). The relaxation time of fresh water is about 8.28 ps at  $T = 25^\circ\text{C}$  and 17.57 ps at  $T = 0^\circ\text{C}$  (Ellison, 2007), but since the salt contribution is considered, interpolation functions should be used. The complex relative permittivity of Enceladus' salt water ocean is then given by (cf. Eq. B.26 & B.27):

$$\varepsilon_r^{\text{Enc}} = 85.355 - i \cdot 90.884 \quad (3.73)$$

if the upper estimate of salinity ( $S = 30$  ppt) is considered which will be used in the simulation of this thesis. For a detailed derivation see the appendix B.

## 3.4 Ices and Subsurface Salt Water Ocean on Enceladus

In the following section, different types of terrestrial ices will be discussed, weighing if they could occur on Enceladus, as well as the chemical composition and dielectric properties of its possible existing ocean.

### 3.4.1 Terrestrial and Nonterrestrial Water Ice

Three different types of ices have been seen on Earth yet: meteoric ice, sea ice and marine ice. The type of the ice has an impact on the radar penetration depth and therefore it is important to study possible scenarios on Enceladus. On Earth, as well as on Enceladus, the ice is composed of polycrystalline Ice  $I_h$ . These ices form ice bodies of different origin, location and extension (e.g. ice sheets, ice shelves, ice

caps, glaciers, etc.) which have different structures and impurity contents.

### **Sea Ice**

This type of ice is formed by freezing of ocean water close to the atmospheric interface (Moore, 2000). On the supercooled water surface some small and needle-shaped ice crystals (so-called frazil or slush ice) accumulate to a mushy, oily and congealing layer (so-called grease ice). Its structure has several phases (gas, liquid brines, solid salts and other impurities) and therefore sea ice has a relatively high salinity (12 – 15‰). Of the dissolved salts, NaCl makes up the largest proportion (Pettinelli et al., 2015).

### **Meteoric Ice**

Meteoric ice is produced by atmospheric precipitation (snow) that forms glaciers and ice sheets (Moore, 2000). Snow is a porous medium of air, ice crystals and small amounts of chemical impurities. This precipitation of snow flakes is followed by a densification process due to snow accumulation and gravity and transforms snow into firn. The latter is a terminus for compacted snow that underlies freshly fallen snow and is by definition at least one year old. It has a higher density dependence on the compaction strength ( $\simeq 830 \text{ kg m}^{-3}$ ). The compaction seals the water- or air-filled connecting paths, forming polycrystalline ice that only has individual bubbles and affects the size of the ice crystals. This is pressure-dependent, so that in glacier ice bubbles only exist above a critical depth (Pettinelli et al., 2015).

### **Marine Ice**

The existence of marine ice is a special case of ice found on Earth, beneath the Amery Ice Shelf in Antarctica. It forms from frazil ice crystals directly in the ocean water where the ice platelets accumulate at the ice shelf bottom, building a mushy ice layer and leading to a compaction similar to the firn densification process of meteoric ice. Small grains lie at the top of the older, upper layer, whose conductivity is about 100 times lower than the meltwater at the bottom and is mainly free of air bubbles. On the other hand, the lower part which is hydraulically connected to the ocean water is highly porous. Marine ice has a high salinity and therefore attenuates radio waves to a greater extent.

### **Discussion**

According to Moore (2000) and other authors, sea and meteoric ice could not be present on Europa due to the lack of an atmosphere of Europa (as well as of Enceladus), since they are formed at the ocean-atmosphere interface and ice formation and metamorphism occurs under different physical and chemical conditions in comparison to Earth (Pettinelli et al., 2015). The presence of marine ice or conceptually similar forms of it, on the other hand, is seen as a possible scenario, since processes that run on Earth could occur at the ice-ocean interface.

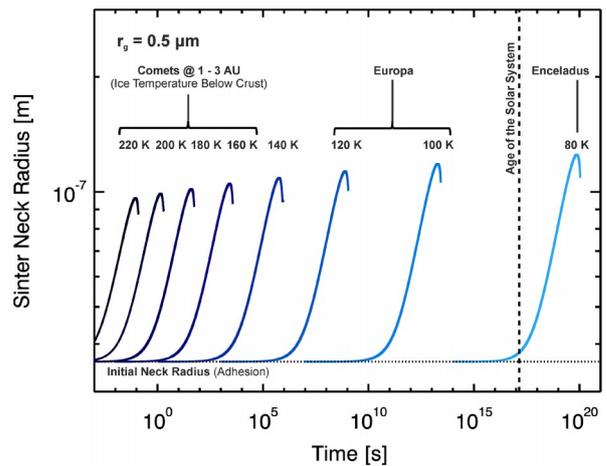
**Special Case: Sintering Processes and Sintered Water Ice Particles on Icy Moons**

Sintering is a temperature-dependent metamorphism process where two neighbouring particles nestle together by forming a sinter neck (grain-to-grain contact area) between them. This formation is due to the need of the system to minimize its surface energy which leads to mass transport into the neck region that enlarges temporally. Its study is important in order to survey the (thermal) evolution of icy planetary surfaces. To sum up, the material undergoes a densification (compactification) process at sufficiently high temperatures and experiences grain growths. For water ice particles (snow) the dominant transport process is the transport of water vapor, i.e. the sublimation of water-ice molecules from the surface and their recondensation in the neck region. Enceladus produces micrometer-sized water



**Figure 3.3:** Schematic of a sintered particle (a). Two particles stick together at the grain boundary (sinter neck). Sinter neck formation of bronze particles in a scanning electron micrograph (b) (German, 2014).

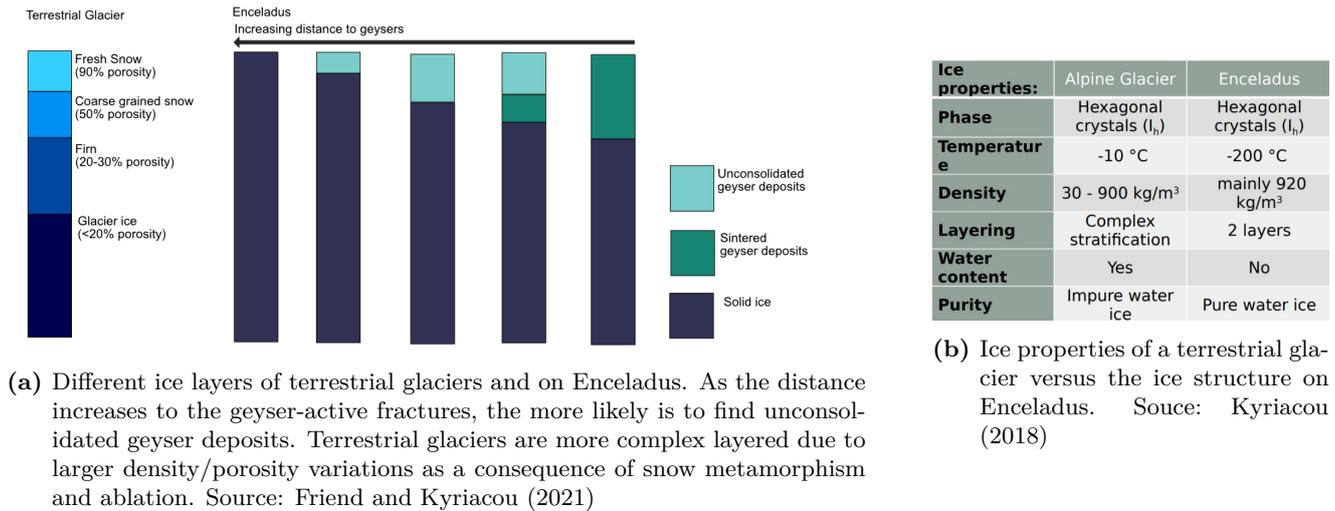
ice particles through its cyrovolcanically plumes. The larger particles fall back onto surface, stratifying it in a layer consisting of snow-like granular powder. The icy moon has an elevated temperature on the SPT near the Tiger Stripes. It is therefore legitimate to scrutinize if sintering process could occur there. According to Gundlach et al. (2018), a temperature of 80 K is still not sufficient for sintered ice to occur since it requires a formation time that is longer than the age of the Solar System. Molaro et al. (2019) emphasize that many surfaces in the Solar System where sintering could be relevant contain salts such as NaCl (e.g. Jovian moon Europa and Enceladus). The abundance of salts can have a significant effect on the sintering timescales and reduces the ice melting temperature and thus increases the sintering rate. On Enceladus, a 0.1 μm grain has a sinter timescale of ~ 10<sup>3</sup>, but with an abundance of 2% of NaCl per mass it would lower the melting temperature to 270 K. For pure water ice, this means an increase of the surface temperature up to 81 K and a decrease of the timescale to 10<sup>2</sup> yr. For larger grains that exist near the Tiger Stripes this effect is smaller. However, for 1 μm particles an eutectic mixture of water and NaCl would lower the melting temperature down to 253 K which means an increase of the sinter timescale from above to below the age (cf. Fig. 3.4) of the Solar System (Molaro et al., 2019).



**Figure 3.4:** Sintering timescales for different temperatures and celestial bodies in the Solar System (Gundlach et al., 2018)

### 3.4.2 Stratification of Terrestrial Glaciers and on Enceladus

Glaciers on Earth are more complex layered than the ice structure on Enceladus is (cf. Fig. 3.5a). The ice on Enceladus is much colder (cf. Fig. 3.5b) and in contrast to the terrestrial glaciers it is unlikely that they have water content. Furthermore, the Enceladan ice is much purer ( $\sigma < 1 \mu\text{Sm}^{-1}$ ) than the ice terrestrial glaciers. Except of a possible existing sintered snow and a granular ice layer, the ice structure on Enceladus can be regarded as largely homogeneous. Hereby, the probability of a sintered snow layer is greater in vicinity to the geysers because of the higher heat output (cf. Fig. 3.5b).

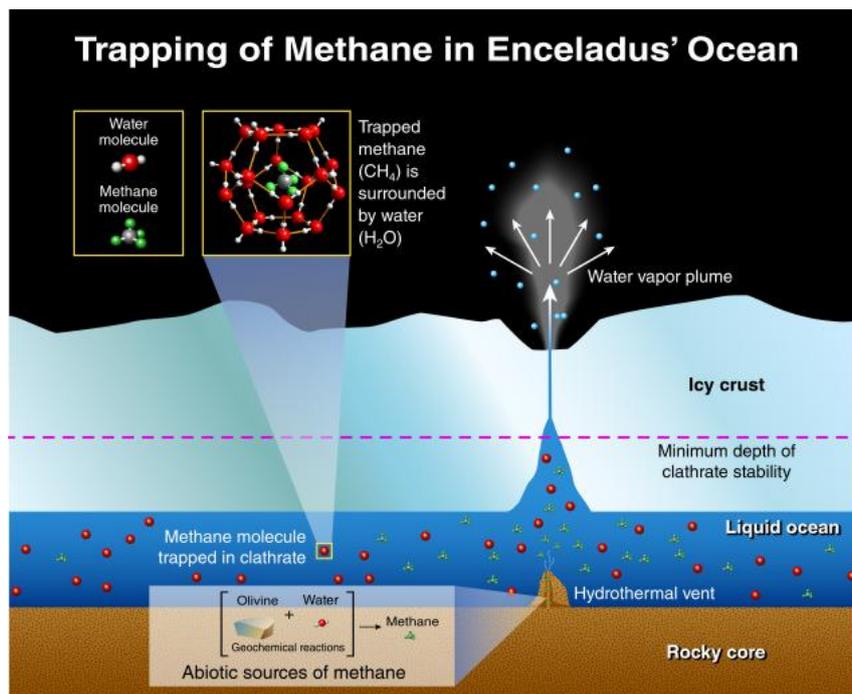


**Figure 3.5:** Stratification of Terrestrial Glaciers and on Enceladus

### 3.4.3 Terrestrial Seawater and Subsurface Salt Water Ocean on Enceladus

Seawater, that is, water with dissolved salts from a sea or an ocean, on Earth has an average salinity of about 35 ppt and a range of pH between 7.9 and 8.25 (average of 8.08) (Raven et al., 2005) to a depth of 50 m, is therefore slightly basic. The most frequently occurring salt is by far sodium chloride ( $\text{Na}^+(\text{aq}) + \text{Cl}^-(\text{aq})$ ) with a salt concentration of 0.47% by mass (molality), followed by magnesium sulfate ( $\text{Mg}^{2+}(\text{aq}) + \text{SO}_4^{2-}(\text{aq})$ ). Most likely the possible subglacial ocean of Enceladus contains dissolved sodium chloride as well and is expected to be 0.05-0.2 molal. The salinity of the ocean is expected to be in a broad range of 4-40 ppt, but for a balanced heat budget that would enable the existence of microbial life organisms, the salinity is indicated to take values between 10 and 30 ppt (Kang et al., 2021), which is only slightly lower than the Earth's average. However, there might exist a discrepancy in the pH. Since direct measurements could not have been done yet, the pH of Enceladus is unknown, but various models have been presented in the past. Postberg et al. (2009) has identified sodium carbonate ( $\text{Na}_2\text{CO}_3$ ) and/or sodium bicarbonate ( $\text{NaHCO}_3$ ) from which he estimated a pH range of 8.5-10.5. However, this is difficult to reconcile with the substantial amount of  $\text{CO}_2$  found in the plume gas. Another approach using a thermodynamic model of carbonate speciation at the conditions of Enceladus' sub-environmental life was suggested in order to determine the pH (Glein et al., 2015). By combining the chemical data of the mass spectra from CDA, which identifies the abundance of  $\text{NaCl}$  and  $\text{NaHCO}_3/\text{Na}_2\text{CO}_3$  in the

salty and basic plume particles, and of INMS, which outlines a significant amount of  $\text{CO}_2$  in the water vapor dominated plume gas, a pH range for the ocean of 11-12 could be derived. Enceladean ocean water would be thus in a clearly alkaline range (corresponds to an ammonia solution) and similar to soda lakes in Earth (e.g. Mono-Lake in California) where multifaceted eco-systems have been found. In total, this model including chemical equilibrium calculations, predicts a  $\text{Na-Cl-CO}_3$  water (Glein et al., 2015). A high pH of Enceladus' subsurface ocean leaves a lot of room for speculation in terms of its cause(s). Nonetheless, knowledge about the pH is of huge geochemical and astrobiological interest, since it can tell whether the environment is habitable for life forms or not. A possible explanation is a process called **serpentinization** where the sea floor of Enceladus' ocean interacts with the chondritic/ultramafic rock of the core. Thereby, olivine containing rocks react to rocks with serpentinite minerals, a process that is well-known from Earth, where iron- and magnesium rich minerals emerge from the sea floor and are washed around by the water, which subsequently becomes basic. Interestingly, molecular hydrogen is produced from serpentinization processes. This reaction is also related to the methane formation already discussed above ( $\text{CO}_2 + 4\text{H}_2 \rightarrow \text{CH}_4 + \text{H}_2\text{O}$ , cf. Fig. 3.6). It could drive both biotic and abiotic synthesis of organic compounds and therefore act as a chemical energy source for life (McKay et al., 2008). Additionally, amino acids can be generated, which are a sort of building blocks for life and nutrition for microorganisms. Yet it remains unclear, if such processes are still ongoing or already extinct, but a direct detection of native hydrogen can provide clarification about that (Affholder et al., 2021).



**Figure 3.6:** A possible explanation for the abundance of methane.  $\text{CH}_4$ -molecules are formed due to geochemical reactions and emitted by hydrothermal vents: in this case, the water circulating at the ground of the ocean is involved in geochemical reactions with the rocky core, where olivine and water react to methane, which could explain the small amount of methane in the plumes. The trapped methane ( $\text{CH}_4$ ) is then surrounded by water ( $\text{H}_2\text{O}$ ) in the ocean, also known as clathrate (in chemistry a clathrate is an inclusion compound of two chemical substances, of which a guest molecule is embedded in a lattice or cage of a host molecule). There might be more complex reactions which induce heavy organic molecules. Source: NASA/JPL (2015)

# Methods of Simulating the Radio Wave Propagation through a Dielectric Medium

## 4.1 Motivation and Goals

Since Finite-Difference Time-Domain (FDTD) methods are full-field solutions, they turn out to be computationally highly expensive. On the other hand, Ray Tracing (RT) is too simplistic and does not account for wave-like phenomena. Both methods will be discussed later. At this point, the methods using the Parabolic Equation (PE) for in-ice radio propagation may be investigated in more detail, which are computationally more efficient than FDTD and include diffractive effects in contrast to RT.

## 4.2 The Parabolic Equation (PE)

The PE is a second-order, linear, constant-coefficient partial differential equation (PDE) and is an approximation of Maxwell's wave equations solutions. It models the energy propagating in a cone centered on a preferred direction, which is called paraxial direction (and will be the positive  $x$ -direction here). These types of PDE's can be hyperbolic, elliptic or parabolic depending on the coefficients  $a$ ,  $b$ , and  $c$  (Apaydin and Sevgi, 2017). The implicit equation of a parabola is an irreducible polynomial of degree two:

$$ax^2 + bxy + cy^2 + dx + ey + f = 0, \quad \text{such that} \quad b^2 - 4ac = 0. \quad (4.1)$$

This is an analytical geometry equation and has exactly the same form as the PE of a function  $u = u(x, y)$ :

$$A \partial_x^2 u + 2B \partial_{xy} u + C \partial_y^2 u + D \partial_x u + E \partial_y u + F = 0, \quad \text{such that} \quad B^2 - 4AC = 0. \quad (4.2)$$

This is where the terminology *parabolic* equation comes from. The PE will first be solved in a domain, where the refractive index  $n = n(x, z)$  has only minor variations from that of vacuum (i.e.  $n = 1 + \delta n$ ) and the boundaries at the top and on the bottom are assumed to be flat (PE-methods have a lack in boundary modelling, this means that for more complex boundary conditions, additional methods should be consulted). In that case, the wave equation can be treated with Fourier transform techniques, which simplifies the solution process. If more strongly varying refractive index profiles are present, the situation is more complex. For that purpose, Hardin and Tappert (1973) presented a split-step/Fourier method where the propagation through the vacuum is superseded with a sequence of phase screens.

### 4.2.1 Derivation of the Wide-Angle Parabolic Equation (WAPE)

Let us start with an arbitrary field  $\psi$ , which is polarized along the  $y$ -axis in a Cartesian system, respectively  $\Psi$ , which is polarized along the angle  $\theta$  in a cylindrically symmetrical system. For horizontal polarization the only non-zero component of the electric field  $\vec{E}$  is this direction of polarization ( $E_y / E_\theta$ ). On the other hand, for vertical polarization the magnetic field  $\vec{H}$  has the only non-zero component in these directions ( $H_y / H_\theta$ ). The scalar wave equation with an  $e^{i\omega t}$ -time dependence for a wave propagating in a homogeneous medium with refractive index  $n$  is  $(\vec{\nabla}^2 - \frac{1}{v^2}\partial_t^2)\psi = 0$ . For  $k_0 = \omega_0/c$  ( $k_0$  is a reference wavenumber that is allocated to the wavenumber  $k$  at the depth of the source) and  $v = c/n$  the propagation velocity where  $c$  is the vacuum speed of light this can be rewritten as:

$$\frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial z^2} + k_0^2 n^2 = 0 \quad \text{Cartesian } (x, y, z) \quad (4.3)$$

$$\frac{\partial^2\Psi}{\partial x^2} + \frac{1}{x}\frac{\partial\Psi}{\partial x} + \frac{\partial^2\Psi}{\partial z^2} + k_0^2 n^2 = 0 \quad \text{Cylindrical } (x, \theta, z)^1 \quad (4.4)$$

In the following, only the cylindrically symmetrical case is considered, which is useful for many application examples in the radar range. Let

$$u = \sqrt{k_0 x} e^{-ikx} \Psi \quad (4.5)$$

so that insertion into 4.4 yields

$$\partial_x^2 u + 2ik_0 \partial_x u + \partial_z^2 u + k_0(n^2 - 1)u = -\frac{1}{4x^2}u \quad (4.6)$$

and for dimensionless coordinates  $\xi = k_0 x$ ,  $\zeta = k_0 z$ :

$$\partial_\xi^2 + 2i\partial_\xi u + \partial_\zeta^2 u + k_0(n^2 - 1)u = -\frac{1}{4\xi^2}u \quad (4.7)$$

where  $u = u(x, z)$  is called *reduced function*. In far-field approximation, i.e. for large  $k_0 x \gg 1$ , which is allowed for problems of interest in radio propagation applications, the r.h.s of Equation 4.7 approaches zero and eliminates the singularity produced by the cylindrical source. Factoring 4.4 and applying the far-field approximation  $x \gg 0$  leads to:

$$\left[ (\partial_x + ik(1 - Q))(\partial_x + ik(1 + Q)) \right] u = 0 \quad (4.8)$$

where  $Q = \sqrt{\frac{1}{k_0^2}\partial_z^2 + n^2}$  is a pseudo-differential operator. This is only true for a homogeneous medium with constant  $n$ . If the refractive index varies with range  $x$ , than the operator  $Q$  does not commute with the range derivative operator  $\partial_x$ , i.e.  $[Q, \partial_x] \neq 0$ , and the factorization is not correct. Therefore, caution is advised for applications, where this error should be kept as small as possible. The solutions of the PDE in Equation 4.8 will be in a splitted and solved simultaneously in the form of two paraxial terms

$$u(x, z) = u_+(x, z) + u_-(x, z) \quad (4.9a)$$

<sup>1</sup>Note:  $x_{\text{cartesian}} \neq x_{\text{cylindrical}}$ . The latter notation is commonly used in radar literature instead of  $\rho$ .

with

$$\partial_x u_{\pm} = -ik_0(1 - Q)u_{\pm} \quad (4.9b)$$

where  $\pm$  corresponds to forward respectively backwards propagating waves, that is, these are the PE for outgoing and incoming waves. In the following, the solution for forward propagating fields shall be considered ( $u_+ \equiv u$ ). The solution

$$u(x) = e^{ik_0x(-1+Q)} \quad (4.10)$$

Since the PE-solver is a split-step solver, and thus, we want to calculate the field for the next position ( $x + \Delta x$ ) in each case, Equation 4.10 changes to

$$u(x + \Delta x, z) = e^{ik_0(x+\Delta x)(-1+Q)} = e^{ik_0\Delta x(-1+Q)}u(x, z) \quad (4.11)$$

From this expression it can easily be seen that the solution of  $u(x + \Delta x)$ , thus, the forwards propagating field at a given range, only depends on the previous solution or range  $u(x)$ , significantly reducing the computational time of the code compared to the elliptic wave equation, which has to be solved simultaneously at all points in the integration domain. Since  $Q$  is a non-local and non-analytical operator, in order to find numerical solutions of Equation 4.9b, take the lowest-order Taylor expansion of the square-root and exponential functions. It will return a quite good approximation for small values of  $n$  that change slowly with the height  $z$  and for a narrow range of angles within a few degrees. This does not work for all problems (e.g. in-water acoustic propagation). The expansion results in a simple form of the PE, which is called the standard parabolic equation (SPE) and is a second-order, linear PDE:

$$\partial_z^2 u + 2ik\partial_x u + k_0^2(n^2 - 1)u = 0 \quad (4.12)$$

and is, as above-mentioned, a narrow-angle approximation of the parabolic wave equation. Since the first term in the Taylor expansion is  $\frac{1}{k^2}|\frac{\partial^2 u}{\partial z^2}| = \sin^2 \alpha$ , the error is proportional to  $\sin^4 \alpha$ . In concrete: it is  $10^{-7}$  for an angle of  $1^\circ$ ,  $10^{-3}$  for  $10^\circ$  and  $10^{-2}$  for  $20^\circ$ . Feit and Fleck (1978) find an expression for the operator  $Q$  that can be applied for simulations considering wider angles and for more strongly varying  $n$ . It has the same lowest-order expansion as the SPE but differs at higher order. If  $Q = \sqrt{\frac{1}{k_0^2}\partial_z^2 + n^2}$  is written as

$$Q = \sqrt{1 + A + B} \quad (4.13)$$

then with the assumption that the operator can be split into (Guan et al., 2018):

$$Q = \sqrt{1 + A + B} \sim \sqrt{1 + A} + \sqrt{1 + B} - 1 \quad (4.14)$$

where

$$A = \frac{1}{k^2}\partial_z^2 \quad B = n^2 - 1 \quad (4.15)$$

the expression can be subdivided into a refractive and diffractive part:

$$Q_{FF} \sim \underbrace{\sqrt{1 + \frac{\partial_z^2}{k_0^2}}}_{\text{diffractive}} + \underbrace{n - 1}_{\text{refractive}} \quad (4.16)$$

with an error of  $\Delta Q = \mathcal{O}(AB)$ . Finally, the Feit-Fleck WAPE is obtained through insertion into Equation 4.9b:

$$\frac{\partial u}{\partial x} = ik \left( \sqrt{1 + \frac{1}{k_0^2} \frac{\partial^2}{\partial z^2}} - 1 \right) + ik_0(n-1)u \quad (4.17)$$

This approach is suitable when  $n \sim 1$ , which corresponds to in-air-radio or in-water-acoustic cases (Prohira et al., 2021). This is not the case for in-ice-scenarios, thus, a different approximation for  $Q$  is to be used:

$$Q_{ice} \sim \sqrt{1 + \frac{\partial_z^2}{k_0^2}} + n \sqrt{1 + \frac{1}{n_0^2}} - \sqrt{1 + \frac{n^2}{n_0^2}} \quad (4.18)$$

It has the the same lowest-order expansion as the SPE and differs at high orders as well.

#### 4.2.2 Split-Step Parabolic Equation (SSPE) Fourier Transform Solution

The solution of the PE has to be done numerically. An efficient tool is the Fourier transform: transform the PDE, solve the transformed equation in the spectral domain (which is simpler and more efficient) and transform the solution back to the original domain. This has only be done for the diffractive component, since the refractive part can be solved via simple multiplication. The Fourier transform of  $u(x, z)$  is:

$$U(x, p) = \mathcal{F}(u(x, p)) = \int_{-\infty}^{+\infty} u(x, z) e^{-2\pi i p z} dz \quad (4.19)$$

and the inverse Fourier transform:

$$u(x, z) = \mathcal{F}^{-1}(U(x, p)) = \int_{-\infty}^{+\infty} U(x, p) e^{2\pi i p z} dp \quad (4.20)$$

The inverse Fourier transform in Equation 4.20 simply is the decomposition of the field  $u(x, z)$  on a vertical into an angular spectrum  $U(x, p)$  of plane waves depending on a spectral variable  $p$ . In order to proceed, two identities of the Fourier transform, which is a linear operator and satisfies a derivation rule, should be used:

$$\mathcal{F}(a \cdot f + b \cdot g) = a \cdot \mathcal{F}(f) + b \cdot \mathcal{F}(g) \quad (4.21a)$$

$$\mathcal{F}(D^\alpha f)(\xi) = i^{|\alpha|} \xi^{|\alpha|} \mathcal{F}(f)(\xi), \quad f \in W^{k,2}(\mathbb{R}^n) = H^k(\mathbb{R}^n), \quad |\alpha| \leq k \quad (4.21b)$$

where  $f$  is a  $k$ -fold weakly differentiable  $L^2$ -function in Sobolev space and  $D^\alpha$  is the derivative operator of the  $\alpha$ -th order. Applying Equation 4.21b to this case yields:

$$i) \mathcal{F}(\partial_z^2 u) = -k_z^2 \mathcal{F}(u) \quad ii) \mathcal{F}(\partial_x u) = \partial_x \mathcal{F}(u)$$

Hence, together with the identity 4.21a, the Fourier transform  $\mathcal{F}$  of the SPE (Eq. 4.12) leads to:

$$\partial_x U = \frac{ik_0}{2} \left[ (n^2 - 1) - \frac{k_z^2}{k_0^2} \right] U \quad (4.22)$$

where  $U(x, k_z) = \mathcal{F}(u(x, z))$  and  $k_z$  is now the spectral variable. Analogously to Equations 4.9b and 4.10, this differential equation can be solved using the exponential function

$$U(x, k_z) = \exp\left[\frac{ik_0}{2}\left((n^2 - 1) - \frac{k_z^2}{k_0^2}\right)x\right] \quad (4.23)$$

and the split-step solving method:

$$U(x + \Delta x, k_z) = \exp\left[\frac{ik_0\Delta x}{2}\left((n^2 - 1) - \frac{k_z^2}{k_0^2}\right)\right]U(x, k_z) \quad (4.24)$$

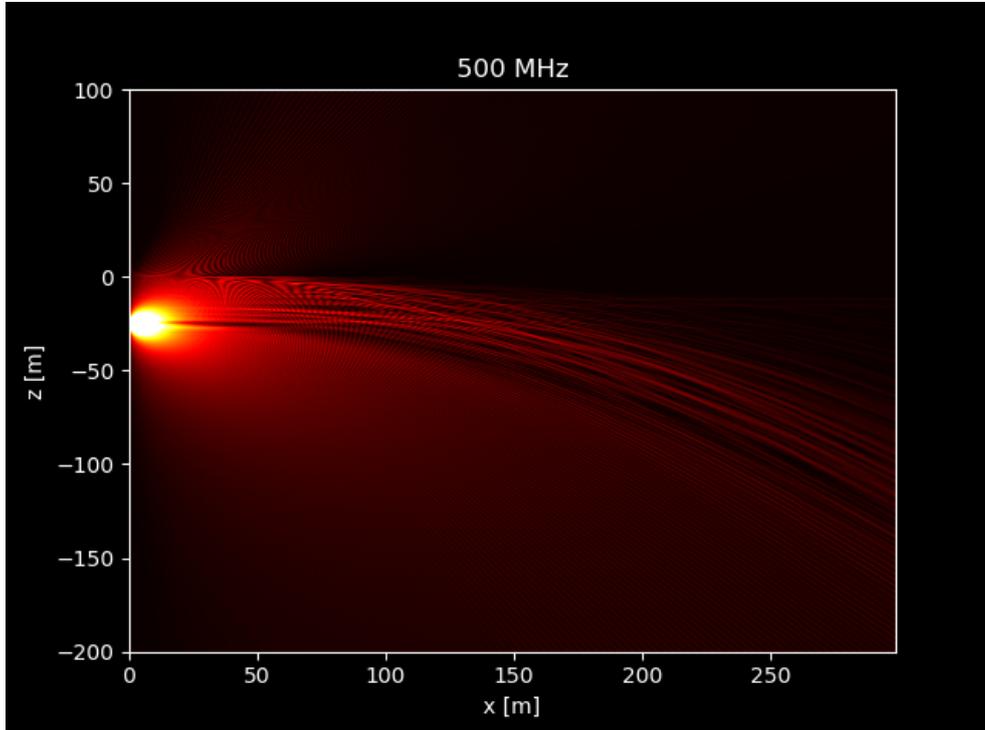
Finally, take the inverse Fourier transform to recompose the solution into the original domain again:

$$u(x + \Delta x, z) = \exp\left[\frac{ik_0\Delta x}{2}(n^2 - 1)\right] \times \mathcal{F}^{-1}\left\{\exp\left[-\frac{i\Delta x}{2}\frac{k_z^2}{k_0}\right]U(x, k_z)\right\} \quad (4.25)$$

This is the split-step solution (SSS) of the standard parabolic equation (SPE). Again, the pseudo-differential operator  $Q$  can be modified to suit to the in-ice scenario with more strongly varying refractive indices and wider angles to  $Q_{\text{ice}}$  (cf. Eq. 4.18). The final SSS of SPE that is implemented in paraPropPython is given by the following expression (Prohira et al., 2021):

$$u(x + \Delta x, z) = \exp\left[ik_0\left(n\sqrt{1 + \frac{1}{n_0^2}} - \sqrt{1 + \frac{n^2}{n_0^2}}\right)\Delta x\right] \times \mathcal{F}^{-1}\left\{\exp\left[-ik_0\Delta x\sqrt{1 - \frac{k_z^2}{k_0^2}} + 1\right]U(x, k_z)\right\} \quad (4.26)$$

An example of a PE-simulation can be seen in Figure 4.1.



**Figure 4.1:** Simulation of the EM waves in a refractive index profile from Taylor Dome Ice Sheet in Antarctica with random density fluctuations included.

### 4.2.3 Summary and Interpretation

Parabolic Equation methods are an approximation of the full wave equation. In this case it calculates the reduced electric field, which is cylindrically-polarized along  $\theta$  within a certain narrow-angular range in the paraxial propagation direction (i.e. within a cone) in far field. For wider angles with respect to the paraxial direction, and, in order to describe radio propagation in a dielectric medium, PE can provide stepping solutions, since it is a split-step solver. This means, in order to calculate the electric field at some distance from the source  $x + \Delta x$ , only the electric field at the previous step  $x$  is required (cf. Eq. 4.11). Thereby, each step in the spatial domain is split in a diffractive and refractive component. For the time-domain, similar Fourier transform techniques can be applied (Prohira et al., 2021).

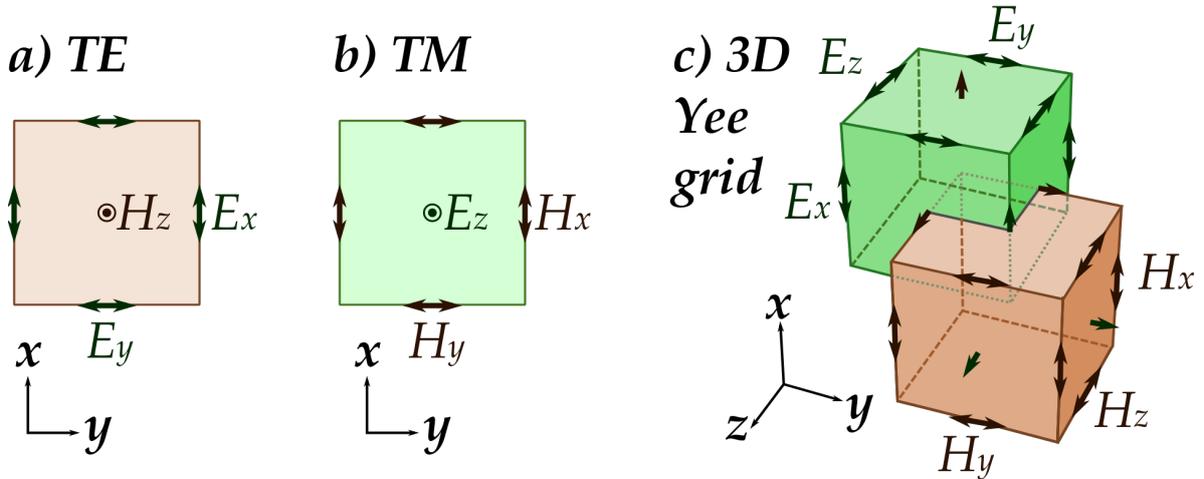
## 4.3 Finite-Difference Time-Domain (FDTD)

Finite-Difference Time-Domain or Yee's method (named after the mathematician Kane S. YEE) is a grid-based numerical modeling method in order to integrate time-dependent differential equations and is therefore predominately applied to solve Maxwell's equations. The basic idea of Finite-Difference methods (FDM) is to approximate the spatial derivatives in the differential equation at finitely many equidistant grid points by central difference approximation. For example, in vacuum it is (Sullivan, 2013):

$$\frac{E_x^{n+1/2}(k) - E_x^{n-1/2}(k)}{\Delta t} = -\frac{1}{\varepsilon_0} \frac{H_y^n(k+1/2) - H_y^n(k-1/2)}{\Delta x}, \quad (4.27)$$

where  $n$  means a time step  $t = \Delta t \cdot n$ . Maxwell's equations consist of such spatial derivatives and time-dependencies: the temporal change of the electric field is given by the spatial change (rotation) of the magnetic field and vice versa (cf. Eq. 3.1 & Eq. 3.2). First, the computational domain should be defined. By means of a so-called Yee-lattice, the space is staggered into several electrical and magnetical grid points (2D) or voxels (3D) and thus the Maxwell's equations become discretized and replaced by a set of finite difference equations (Yee, 1966). Each cell can be specified in terms of its material (free air, dielectric, metal). After that, the source should be defined, which can be current on a wire, an electric field, or a plane wave. The finite-difference equations are solved by a software or in a leapfrog scheme (hardware): The electric field vector components are solved at a given instant in time in a space volume. Then, the magnetic field components are solved in the same volume at a next instant in time and depends on the stored value of the previous magnetic field and the numerical curl of the electric field. This process is iterated repeatedly for all of the grid points in this alternating manner so that wave propagation through a numerical grid stored in computer memory can be simulated. One beneficial point is that by a single simulation the response of a wide range of frequencies can be obtained when using a broadband pulse, since FDTD is a time-domain solver. Disadvantageous, however, is probably the fact that the computational domain must be finite and can only be solved by introducing appropriate boundary conditions such as a „perfectly matched layer“ (PML), which involves errors. Otherwise, the outgoing fields will be reflected back into the problem space and superimpose the desired solution (Sullivan, 2013). Besides, for an accurate solution the gridded domain must be fine enough to grasp the smallest wavelengths and smallest geometrical features (e.g. long, thin wires), leading into an extremely

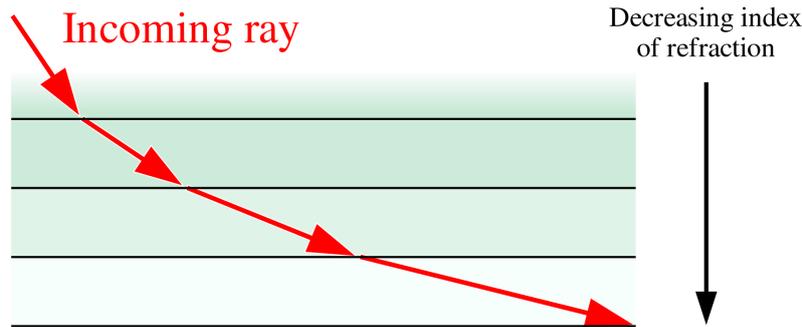
high computational time and large memory requirements. Essentially, FDTD differ from PE methods by the definition of the source and the boundary conditions.



**Figure 4.2:** Staggered lattice (Yee-grid) used in FDTD. Source: wikipedia.org (2015). Credit: Filip Dominec

## 4.4 Ray Tracing (RT)

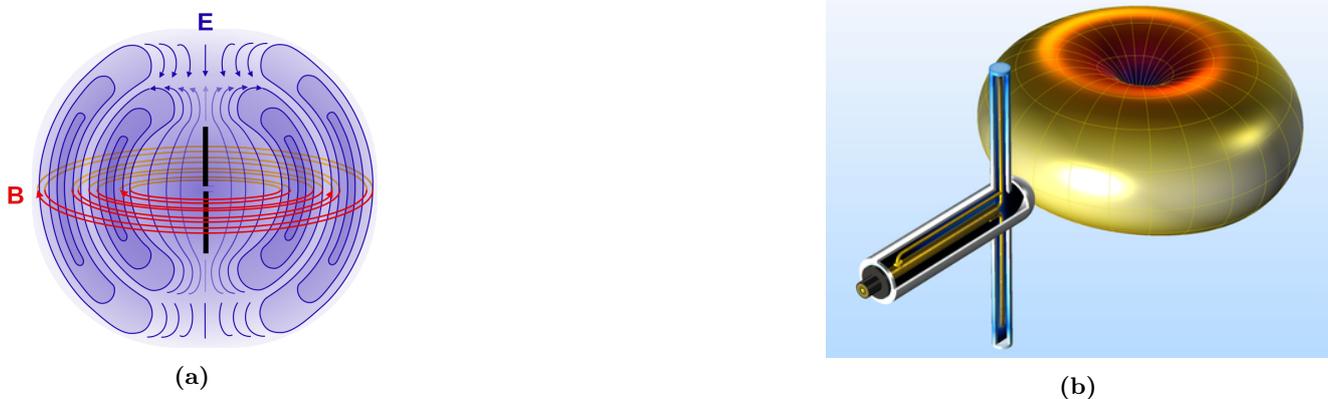
Ray Tracing (not to be confused with ray tracing from 3D computer graphics) is a standard simulation technique in order to calculate the path of waves or particles through an optical medium with changing refractive index, absorption characteristics and reflecting surfaces. It makes use of geometrical optics, where a ray is an idealized model of light: a ray is a straight or curved line that is perpendicular to the light's wavefronts. Hence, a wave can be modeled as a series of many narrow beams (rays) that in a certain small distance can be assumed to be locally straight (Spencer and Murty, 1962). In terms of wave optics, RT takes the infinite frequency limit where the wave changes to a ray. The procedure looks as follows: the incident ray is advanced by a small amount, exactly the small distance where it is straight, through the medium with changing refractive index (Andersen and Kak, 1982). At each step, the new direction of the ray is calculated from which the new ray is sent out (cf. Fig. 4.3). Following this concept, this can be done over and over again until the full path is completed. It can account for direct signals, that is, signals that travel without interceptions with the surface, and reflected signals from two-layer boundaries (for example air/ice). RT algorithms use approximations to Maxwell's equation but is not capable to describe wave phenomena such as interference and diffraction. Consequently, in contrast to wave or field methods, where the field is solved at each point regardless of which path the rays take, RT cannot find solutions in forbidden zones (so-called shadow-zone propagation), caused by total internal reflection. Hence, for in-ice scenarios in order to detect neutrinos for example, RT over-simplifies wave propagation, so that a reconstruction of their incident direction seems to be impracticable (Prohira et al., 2021), and thus gets outperformed by FDTD and PE methods. Notwithstanding, RT can not be replaced for vast simulation geometries due to its super-fast computational time. A 1.5 km x 1.5 km domain can be solved within 30 minutes with PE, but it is still longer than with RT (Prohira et al., 2021).



**Figure 4.3:** Advanced incoming ray that is assumed to be straight within each layer. Source: wikipedia.org (2008). Credit: Vanessa Ezekowitz

## 4.5 Dipole Antennae

Whether idealized or in reality, (transmitting and receiving) antennae are used in radar wave techniques and modeling. antennae are technical arrangements for the radiation and reception of electromagnetic waves. As a transmitting antenna (hereafter denoted as TX) it converts conducted electromagnetic waves into free-space waves, or conversely, as a receiving antenna, it converts electromagnetic waves arriving free-space waves back into conducted electromagnetic waves. A dipole antenna is an antenna with a center-fed driven element for transmitting and receiving radio frequency waves. It consists of a straight electric conductor such as copper, which is interrupted at the center in order to be fed, into two wires that produce the poles (cf. Fig. 4.4a). The most common used dipole antennae are half-wave dipole antennae, i.e. antennae whose total length is about half the wavelength they are driven with. The radiation pattern of an antenna is associated with the directional (angular) dependence of the electric field strength. The field of a dipole is omnidirectional in the plane perpendicular to the wire axis and vanishes to zero on the axis. The radiation pattern of a half-wave dipole has a  $\sin^2 \theta$ -shape and is therefore maximum perpendicular to it and declines to zero on the axis. In three dimensions, the radiation pattern has a toroid or doughnut shape (cf. Fig. 4.4b). By means of a hertzian dipole, which is an idealization an electromagnetic wave transmitting object, the radiation of real transmitting antennae as well as the directivity of real receiving antennae as a measure of the gain, can be calculated.



**Figure 4.4:** Electric (blue) and magnetic (red) field lines transmitted during the radiation of a half-wave dipole antenna (black) (a). Source: wikipedia.org (b). Simulation of the 3D radiation pattern (b). Source: comsol.de

The source implemented in paraPropPython is a vertically polarized dipole source at depth  $z_0$  and halves of  $L = \frac{\lambda}{4}$ . The  $\hat{z}$ -component of the reduced field within this source region is defined by

$$u(0, z_0 - L : z_0 + L) = A[\hat{n} \times \hat{\varepsilon} \times \hat{n}]_z \quad (4.28)$$

where  $A$  is a complex amplitude.  $\hat{n}$  is a unit vector that points out radially from the dipole and  $\hat{\varepsilon} = (0\hat{x}, 0\hat{y}, 1\hat{z})$  is the polarization vector of the antenna (Prohira et al., 2021). For the identification of obstacles the received power of high-gain antennae can be considered. It limits the range of a surface-penetrating radar and is mainly dependent on material loss, spreading loss and target-reflection or scattering loss ( $L_{\text{tot}} = \sum_i L_i$ ). A more precise breakdown of all contributions can be read in Daniels (1996). The power  $P_r$ , which is proportional to the inverted fourth power of distance for a point reflector and can be calculated by FRII's transmission formula:

$$\frac{P_r}{P_t} = \frac{G_t A_r}{(4\pi R^2)^2} \quad (4.29)$$

where  $P_t$  is the transmitted power,  $G$  is the gain of the TX and  $A_r$  is the receiving aperture and  $R$  is the range to the target. The spreading loss can be expressed as

$$L_s = 10 \log_{10} \frac{P_r}{P_t} = 10 \log_{10} \frac{G_t A_r}{(4\pi R^2)^2} \propto \frac{1}{R^4} \quad (4.30)$$

and the attenuation loss decreases with an exponential function:

$$L_a = e^{-2\alpha R} \quad (4.31)$$

For the identification of a flat water pocket the received power is proportional to the inverted second power of distance:

$$\frac{P_r}{P_t} = \frac{G_t A_r}{4\pi R^2} \propto \frac{1}{R^2} \quad (4.32)$$

The attenuation loss is  $L_a = e^{-\alpha R}$ . Finally, the total loss is proportional to the transmission coefficients of the ice-water boundary layer:

$$L_{\text{tot}} \propto (1 - t_{\text{ice} \rightarrow \text{water}}) \cdot e^{-\alpha R} \quad (4.33)$$

with  $t_{\text{ice} \rightarrow \text{water}}$  defined by Equation 3.51:

$$t_{\text{ice} \rightarrow \text{water}} = \frac{2\sqrt{\varepsilon_{\text{water}}}}{\sqrt{\varepsilon_{\text{water}}} + \sqrt{\varepsilon_{\text{ice}}}} \quad (4.34)$$

## 4.6 Codes

The codes that run the simulations and analyze the A-Scans and the B-Scans can be looked up on GitHub: <https://github.com/boccarella/paraPropPython.git>.

paraPropPython.py and Run-Simulation.py are also attached in the appendix D. The data of the simulations S1-S5 can be requested by contacting me at the following email:

gianluca.boccarella@uni-wuppertal.de

# Radio Wave Simulation in a Realistic Scenario on Enceladus

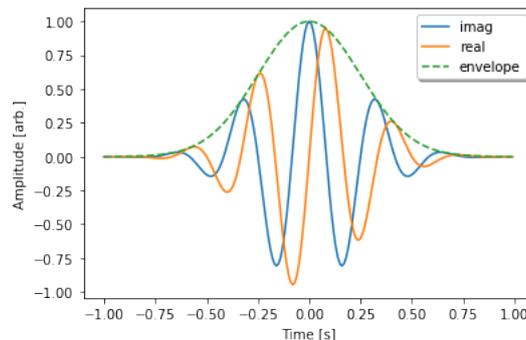
In this chapter, realistic simulation domains on Enceladus, consisting of multiple-layer dielectric profiles will be shown. “Realistic” means within the scope of possibilities available here: from Cassini data, especially from mass spectrometries of CDA that examined the chemical compositions of the plume particles and their deposition rates on the surface, the required dielectric parameters are determined. They are derived in the appendix B.

## 5.1 Methods of Analysis

The aim of this work is to perform simulations of a radar scan of Enceladus’ like ice with a water pocket and then attempt to reconstruct the depth of the water pocket and to prove the scope of application of so-called borehole ground-penetrating radar methods. To be able to do this, a couple of radar techniques are required, which will be presented in the following section.

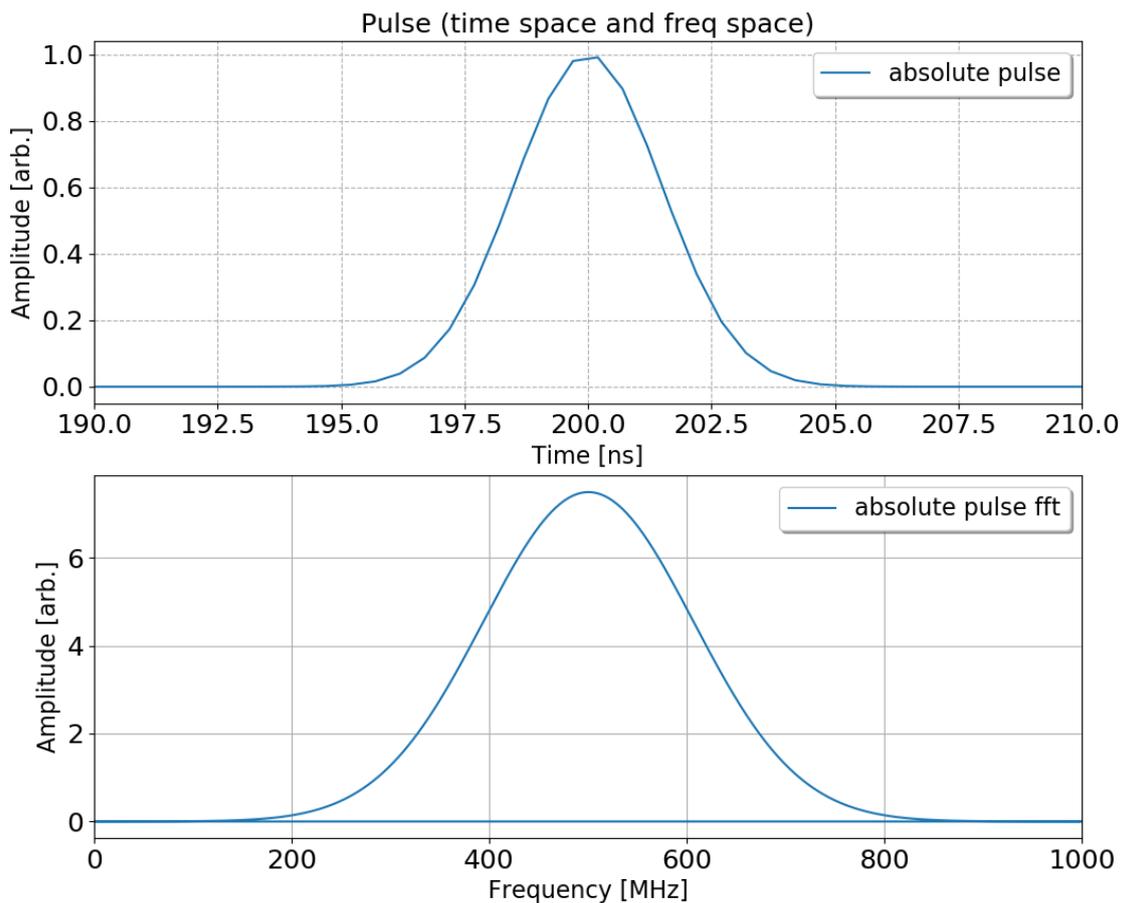
### 5.1.1 A-Scan processing: Simulation of the Propagation Time

In order to find the time (TOF) that both the direct transmitted signal and the reflected signal take during their propagation through the dielectric medium, the following methods of time-domain simulations are a valuable method to derive it. As a first step, a Gaussian pulse (cf. Fig. 5.1) with central frequency of  $f_c = 500$  MHz is created.



**Figure 5.1:** Example of a Gaussian Pulse sampled at 100 Hz for 2 seconds. The blue line is the imaginary component, the orange line is the real component and the green dashed line is the envelope which has the form  $\exp(-a \cdot t^2) \cdot \exp(2\pi i \cdot f_c \cdot t)$ .

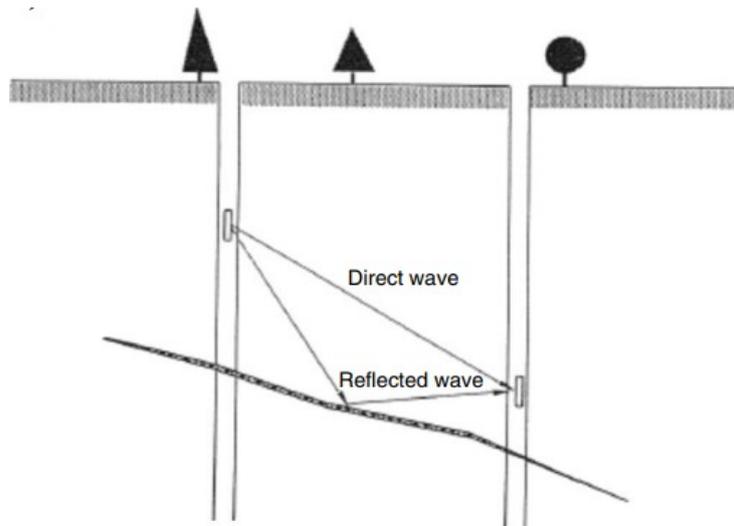
This input signal can be decomposed in its constituent frequencies in a complex spectrum by using a Fourier transform. For reasons of efficiency, a Fast Fourier Transform which is an algorithm used to compute discrete Fourier transforms will be applied (cf. Fig. 5.2). In so doing, the simulation can be run for each single frequency at a each single source depth. Hereby, the dipole amplitude of the source is set as the complex amplitude at each frequency, which will be used to solve the electromagnetic field for a given source depth across the simulation geometry. After propagating through the medium with losing energy (attenuation) and interacting with the boundary layers of the media with different permittivity by scattering (refraction, reflection and diffraction) the complex amplitude of the incident pulses are sampled at the desired RX positions in a received spectrum. The received spectra can be recomposed back into their original state by using the inverse Fourier transform. These receiver signals are also known as *A-Scans*. To filter out unwanted frequency components, a high-pass filter is applied. The propagation time of all reflected and non-reflected signals can be extracted by looking at the time domain of the pulses at each RX position.



**Figure 5.2:** Gaussian Pulse sampled at 500 MHz for 1000 ns with a pulse center of 200 ns and an amplitude of 1.0 (top) FFT of the the pulse that converts from the time domain into the frequency domain (bottom).

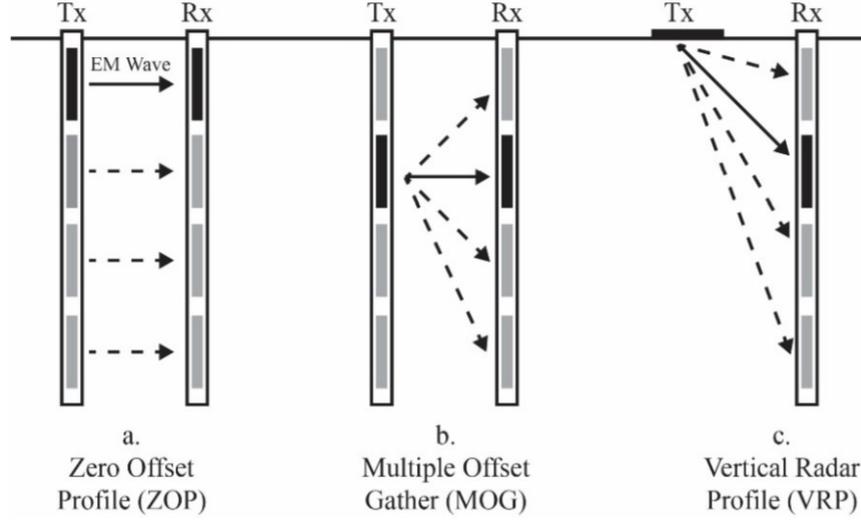
### 5.1.2 B-Scan processing: Borehole Ground-Penetrating Radar

Borehole Ground Penetrating Radar or Borehole Radar (BHR) is a commonly used geophysical mapping tool to identify, localize and characterise buried objects of interest within the subsurface environment of the Earth (Slob et al., 2010) or of other planets like water-bearing crevasses on Enceladus, henceforth denoted as “water pocket”. It can be applied together with SAR by utilizing a so-called transceiver, i.e. a combined transmitting and receiving antenna that moves (“bores”) vertically through a borehole (cf. Fig. 5.3) drilled in the ice. This method including two boreholes is named **crosshole GPR** while two more methods exist using only a single borehole (Slob et al., 2010). At each transceiver depth the time-domain waveform is measured (A-Scan) with the method presented above. By combining the sampled A-Scans for each pair of transmitter and receiver at equal depth relative to each other, a *B-Scan* can be generated. It reproduces the strength of the electric field as a function of depth and time in a color map that can be used with regard to absorptions and reflections. GPR-based characterizations



**Figure 5.3:** Schematic representation of a crosshole GPR measurement in two dimensions. A transmitting and receiving antenna pair bore through the ground in order to measure a planar reflector (Slob et al., 2010).

can be classified into two categories of surveys, namely reflection and transillumination (Craig Jones and Hudyma), that depend on how the antennas are moved and how many boreholes exist. Reflection surveys can be surface reflection surveys where both the TX and the RX are moved over a certain distance in a specified pattern scanning the whole area over the surface and mapping the subsurface, or reflection borehole surveys where both antennae move within a single borehole. Types of transillumination surveys are shown in Figure 5.4. By using the Zero Offset Profiling (ZOP), the antennae are lowered to the same depth in parallel (Craig Jones and Hudyma). This latter principle will be transferred to the simulations.



**Figure 5.4:** GPR-based transillumination surveys. TX and RX can be lowered at the same depth (a), TX is fixed and RX is lowered (b). Both are crosshole configurations. If the TX is located on the surface and the RX is lowered at various depths, one speaks of a hybrid-surface method (Craig Jones and Hudyma).

### 5.1.3 Single-Layer-Scattering

The computation of the water pocket depth is based on simple geometric considerations (cf. Fig. 5.5). The path length  $s$  that the reflected wave will travel can be calculated from the propagation velocity of the wave and the delayed time of its arrival. The velocity of a wave in a dielectric medium is related to the real part of the permittivity (refractive index) of the medium and the speed of light:

$$s = v_{wave} \cdot t_{delayed} = \frac{c}{\sqrt{\epsilon'_r}} \cdot t_{delayed} \iff s^2 = \frac{c^2}{\epsilon'_r} t_{delayed}^2 \quad (5.1)$$

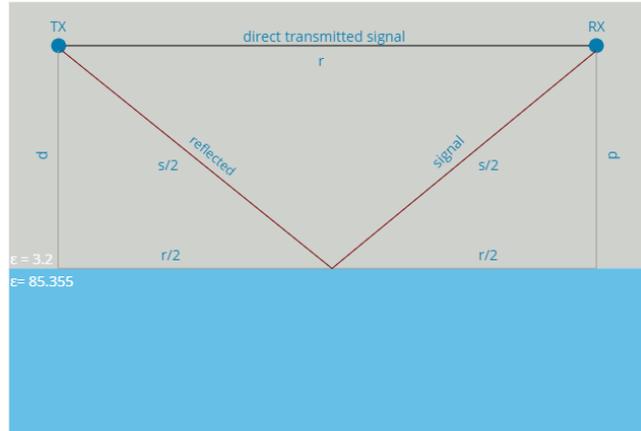
In order to calculate the depth  $d$  where the flat water pocket is located, the Pythagorean theorem will be used, since the triangles are right-angled:

$$d^2 + \left(\frac{r}{2}\right)^2 = \left(\frac{s}{2}\right)^2 \iff d = \sqrt{\frac{s^2}{4} - \frac{r^2}{4}} \quad (5.2)$$

$$d_{water} = \frac{1}{2} \sqrt{\frac{c^2}{\epsilon'_r} t_{delayed}^2 - r^2} \quad (5.3)$$

This gives the distance from the water pocket to the transmitting/receiving antenna. Thus, for the distance from the surface to the water pocket Equation 5.3 should be corrected by the factor of the antenna's actual depth  $d_{antenna}$ :

$$d_{water} = \frac{1}{2} \sqrt{\frac{c^2}{\epsilon'_r} t_{delayed}^2 - r^2} + d_{antenna} \quad (5.4)$$



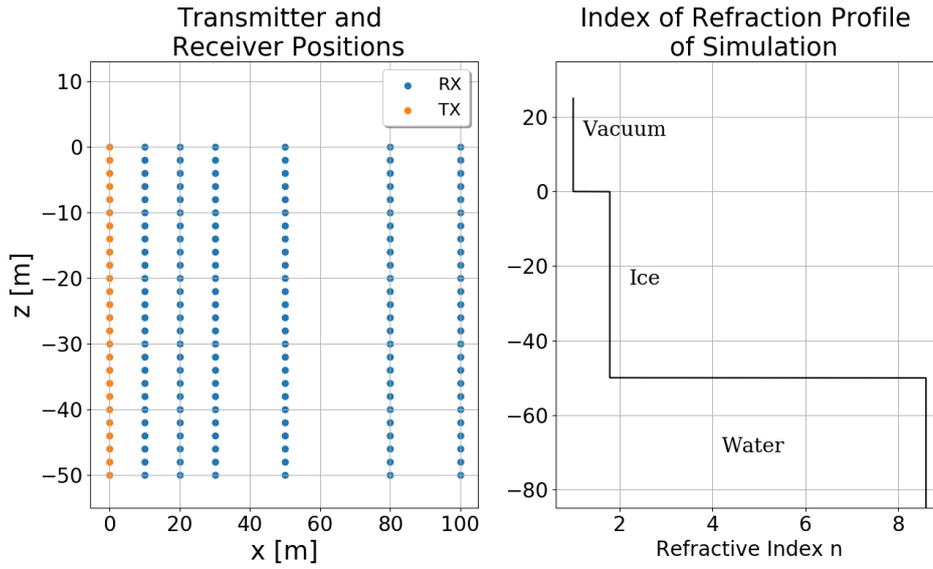
**Figure 5.5:** Sketch for the derivation of the One-Layer-Scattering Algorithm. Hereby  $s$  is the path length for the reflected signal,  $r$  is the range between TX and RX and  $d$  is the depth of the flat water pocket (blue) that has to be calculated. Refraction is not shown in the plot.

## 5.2 Simulation Geometries

To test the accuracy and validity of this simulation method, it will be started with a highly simplified case of pure ice with a density of  $920 \text{ kg/m}^3$  and a water pocket at 50 m of the surface with a flat boundary between the ice and water. The complexity of the domain will be consecutively increased, achieving a sufficiently good approximation of the circumstances on Enceladus in the last simulation. The simulations will be labelled S1 to S5 (cf. Tab. 5.1).

### 1D Refractive Index Profile, Small Scale

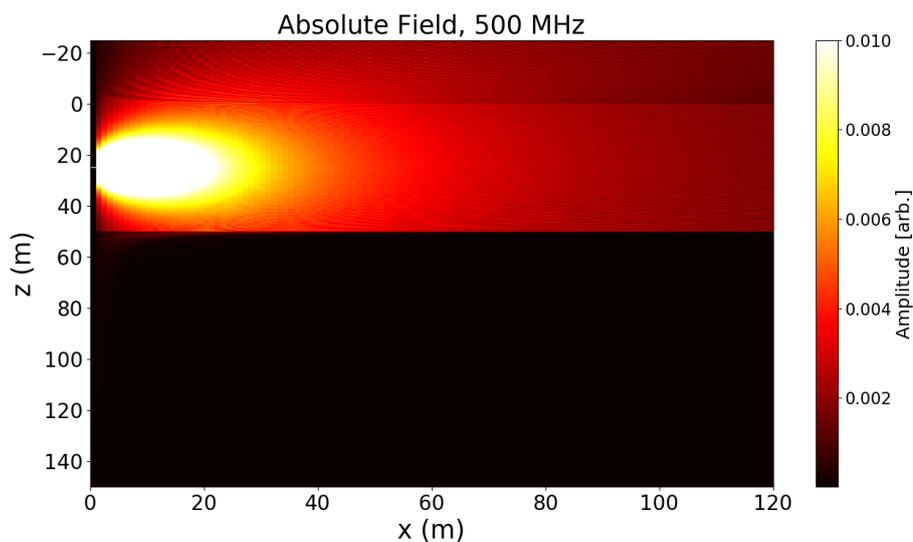
The TX and RX positions as well as the one-dimensional refractive index profile are shown in Figure 5.6 for the 3-layer scenario with pure water ice. The TXs are placed in a range from 0 to 50 m depth separated 2 m apart from each other. The RXs are positioned in an array of [10, 20, 30, 50, 80, 100] m at equal depths to the TXs. The depth of the water pocket is set to 50 meters. The absolute field calculated across the whole simulation geometry is shown in Figure 5.7. The radiation pattern of the half-wave dipole antenna is clearly visible (it is equivalent to Fig. 4.4b in 2 dimensions). Since water is a highly transparent medium at frequencies in the visible spectrum but not at radio frequencies, scattering effects at the boundary layer cannot be recognised well and a logarithmic scale would be required to visualize it. As already discussed above, PE-methods, take the wave character into account unlike RT-methods, which is highly visible in Figure 5.7 based on the lighter and darker areas in the pattern. The refraction at the ice-vacuum boundary layer can be identified by means of the horizontal line at the surface ( $z = 0$  m) and the upwards going beams. The amplitude does not decrease significantly over the range of 120 m as to be expected in the ice (qualitatively from 0.010 at the source to 0.003).



**Figure 5.6:** Positions of the receiving (RX, orange) and transmitting (TX, blue) antennas embedded in the ice with a range of 100 m (left) and one-dimensional, 3-layer refractive index profile (S1) comprising of vacuum, ice and water (right) used for the simulation.

**Table 5.1:** Labels used for the simulations in this work.

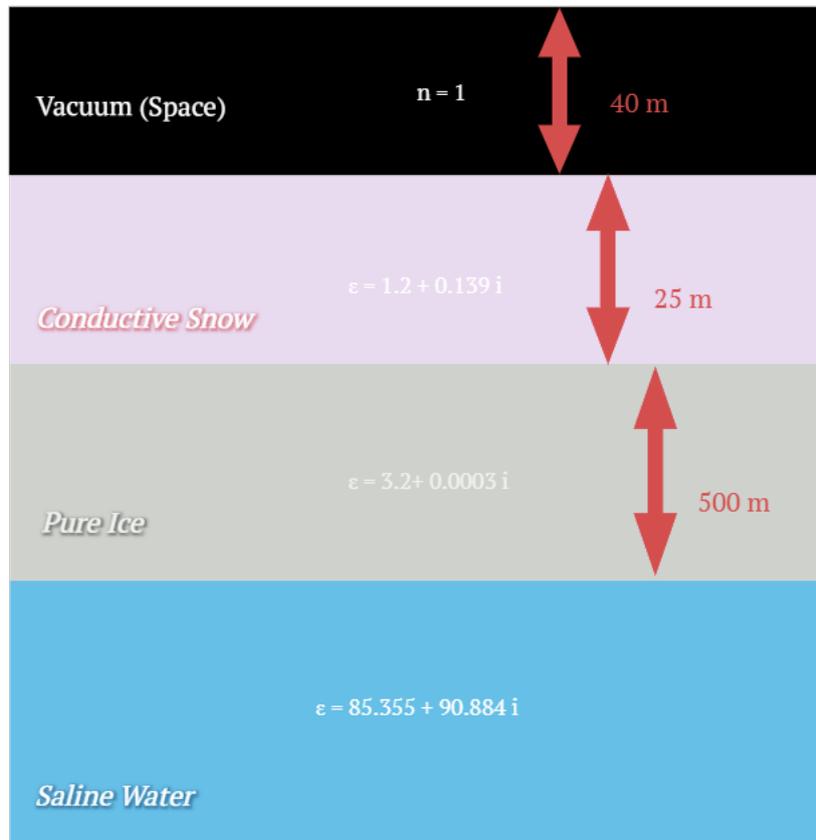
Simulation	Scenario
S1	1D, vacuum + pure ice + flat water
S2	2D, vacuum + conductive snow + pure ice + flat water
S3	2D, vacuum + conductive snow + pure ice + triangular water
S4	2D, vacuum + conductive snow + sintered snow + pure ice + triangular water
S5	2D, vacuum + conductive snow + sintered snow + pure ice + triangular water + meteors



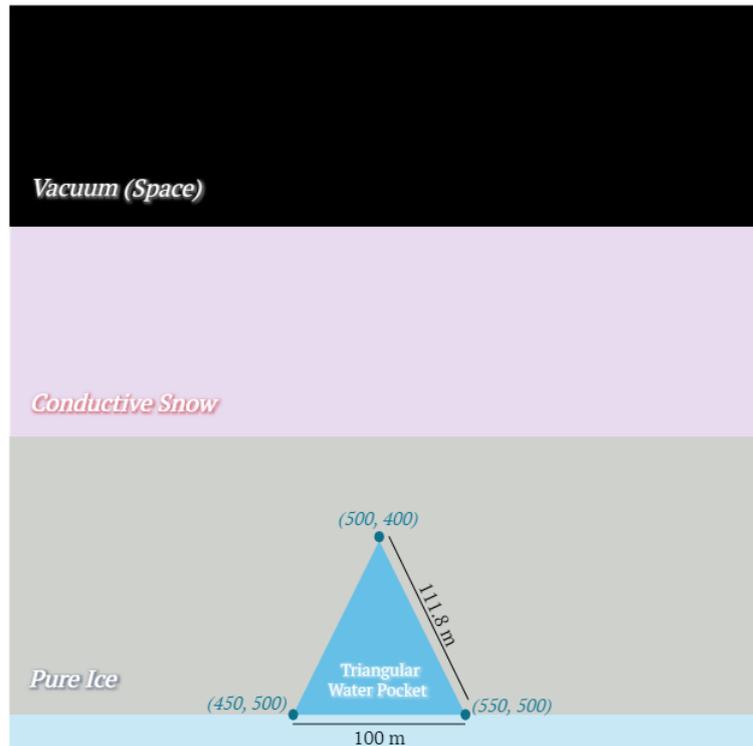
**Figure 5.7:** Absolute field calculated with the split-step solution of the standard parabolic equation with a source at 25 m depth in the ice, a flat water pocket at 50 m and a central frequency of  $f_c = 500$  MHz.

## 2D Refractive Index Profile, Large Scale

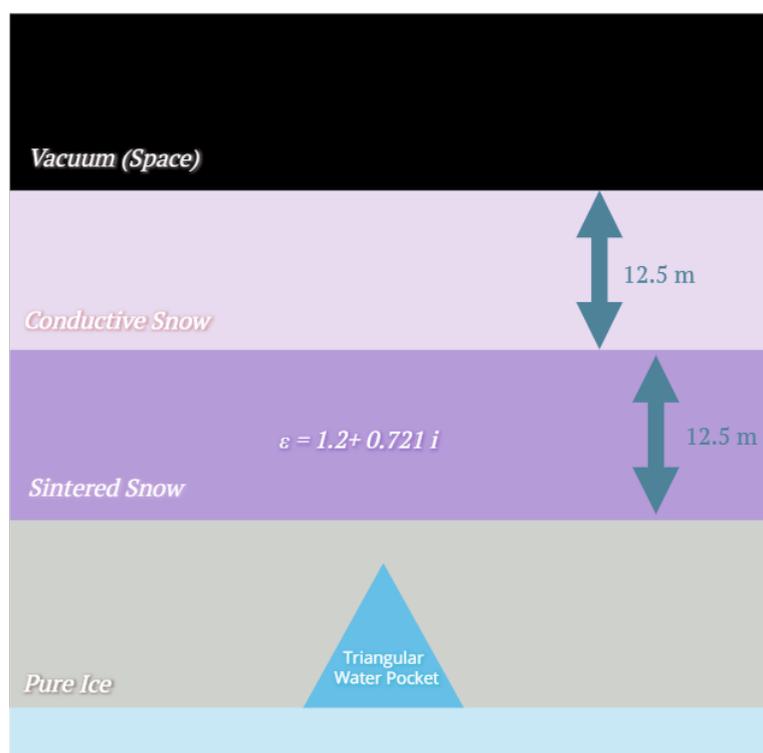
The two-dimensional refractive index profiles of S2-S5 are shown in Fig. 5.8-5.11. The surface of the water pocket might be at 10 % of the way down to the ocean Dotson (2018). Therefore, the 400 meters chosen here are a realistic assumption. Depending on the geyser model, the water-filled area might be triangular-shaped or flat so that the assumptions made here are probably in good agreement with the reality. The ice crust on Enceladus can be assumed to be homogeneous except for the  $\text{CO}_2$  abundance on the SPT in the ice (cf. chap. 3.3.2). For the sake of simplicity, it will be assumed a layer of pure ice. The snow layer is set to 25 m in accordance with geyser particle deposition rate data (cf. Fig. C.1 & C.2). In total 540 RXs are placed throughout the whole simulation domain while 27 of them are placed along each column in the  $z$ -axis in increments of 20 meters starting from -25 m (vacuum) up to 495 m (ice or water) and 20 of them are placed along each row in the  $x$ -direction. The signals are sampled for 12000 ns.



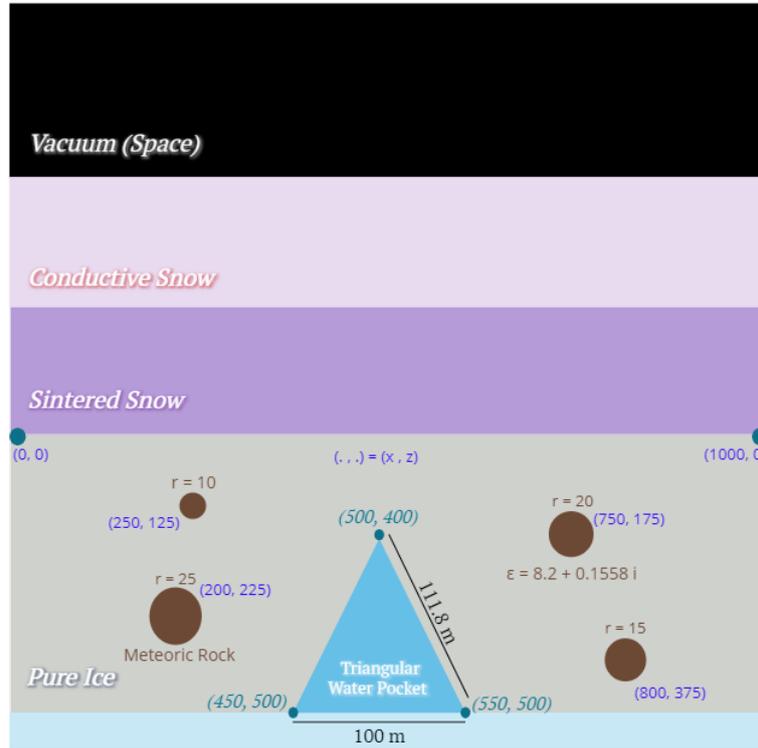
**Figure 5.8:** Multiple layer simulation geometry S2 of 1000 m width with a flat water layer (blue), a 500 m deep pure ice sheet (grey), a 20 m conductive snow layer consisting of sedimented sodium-bearing Type-III particles (pink) and the non-dielectric space (black). This illustration is not to scale.



**Figure 5.9:** Multiple layer simulation geometry S3 with an isosceles, triangular water pocket (blue) at 400 m respectively 500 m with a basis of 100 m length and legs of 111.8 m length. This illustration is not to scale.



**Figure 5.10:** Multiple layer simulation geometry S4 with an isosceles, triangular water pocket (blue) and two layers of consolidated and unconsolidated geyser deposits. This illustration is not to scale.



**Figure 5.11:** Multiple layer simulation geometry S5 with an isosceles, triangular water pocket (blue) and four circular meteoric rocks of different sizes (radii from 10 to 25 m). The position of the center point is indicated in the dark blue brackets and is in meters. The complex permittivity of meteoric rock is determined by Hérique et al. (2018). This illustration is not to scale.

## Permittivities and Refractive Indices

The simulation makes use of different complex permittivities and refractive indices. These were represented and derived before and now compiled in Table 5.2.

**Table 5.2:** Complex permittivities and refractive indices calculated via Eq. 3.30 for the simulations S1-5

Simulation	$\varepsilon_r = \varepsilon' - i \cdot \varepsilon''$	$n^* = n - i \cdot \kappa$
S1	ice: $3.2 - i \cdot 7 \cdot 10^{-5}$	$1.8 - i \cdot 1.9 \cdot 10^{-4}$
	water: $74 - i \cdot 1.8$	$8.6 - i \cdot 0.1$
S2-4	snow: $3.2 - i \cdot 7 \cdot 10^{-5}$	$1.8 - i \cdot 1.9 \cdot 10^{-4}$
	snow: $1.2 - i \cdot 0.139$	$1.097 - i \cdot 0.063$
	saline water: $85.355 - i \cdot 90.884$	$10.248 - i \cdot 4.434$
	meteor: $8.2 - i \cdot 0.1558$	$2.864 - i \cdot 0.027$
	sintered ice: $1.2 - i \cdot 0.721$	$1.800 - i \cdot 0.203$

## Transmission and Reflection Coefficients

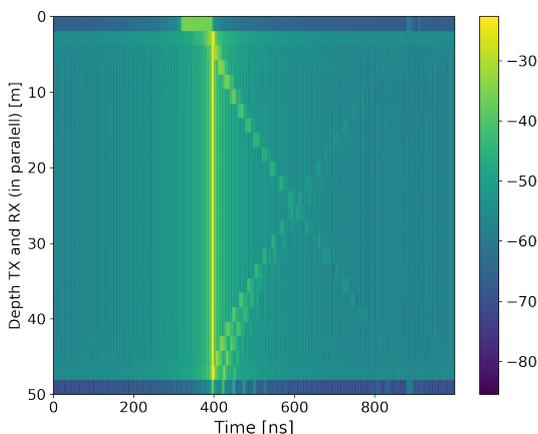
To be able to make statements about the absorption of the radio waves, the reflection and transmission coefficients, that make up a part of the signal loss, are shown in Table 5.3 for all relevant boundary layers. Saline water has the lowest transmission due to the impurities as discussed in chapter 3.3.2.

**Table 5.3:** Transmission ( $t$ ) and Reflection ( $r$ ) Coefficients for the Simulations S1-5 calculated via Eq. 3.51 and the refractive indices from Tab. 5.2.

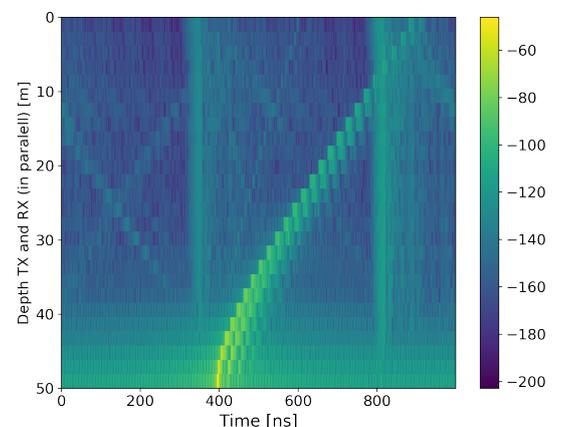
	$r$	$t$
ice-vacuum	0.078	0.922
ice-snow	0.057	0.943
ice-water	0.556	0.444
snow-vacuum	0.012	0.988
snow-sintered snow	0.061	0.939
sintered snow-ice	0.052	0.948
meteors-ice	0.052	0.948
ice-saline water	0.961	0.039

### 5.2.1 Reference Simulation for Signal Filtering

The reflection spectrum is ambiguous due to the large number of reflections, some of which are superimposed on each other. If only the reflection from the water is to be analyzed, the receiver signal must be filtered out by the direct transmitted signal and the reflection from the surface (ice-vacuum interface), as the strength of the water reflection is suppressed, especially in the area near the surface. Figure 5.12a shows the unfiltered signal with the direct pulse and the two reflections (oblique lines). For the water reflection analysis only the oblique line going from the lower left to the upper right is relevant. The rest should be filtered out. This is possible by performing a so-called reference simulation where the simulation geometry only consists of the ice- and the vacuum-layer. By subtracting this reference simulation from the main simulation, only the water reflection (except for small disturbances and inaccuracies of the simulation) will be visible in the spectrum. Figure 5.12b shows the filtered spectrum where the water reflection makes up the largest share.



(a) Unfiltered spectrum in a 3-layer scenario in S1



(b) Filtered spectrum in a 3-layer scenario in S1

**Figure 5.12:** B-Scans at a distance of  $R = 20$  m between TX and RX in S1.

### 5.2.2 Refractive Index of the Simulation

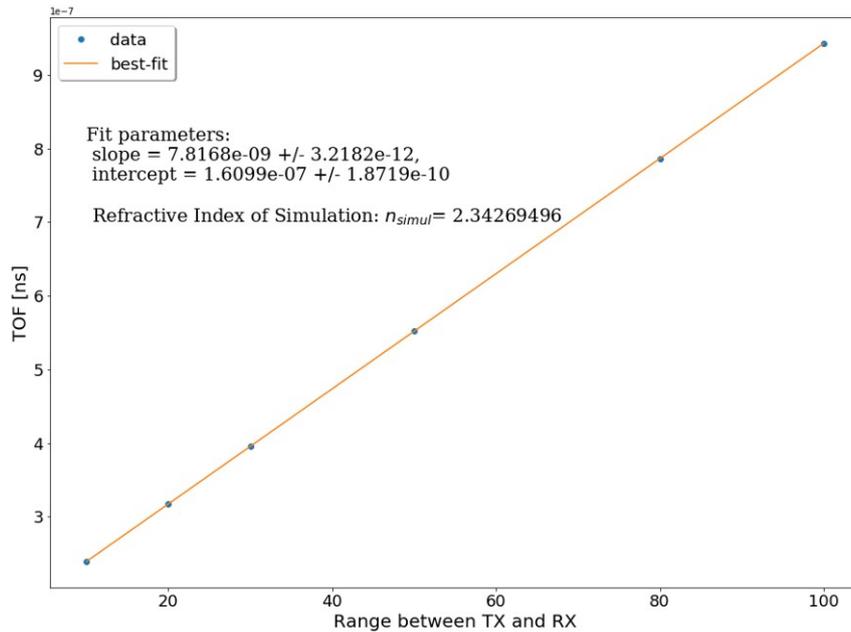
The TOF of the pulse was measured through the medium and it was found that the TOF was larger than expected for the simulation. Thus, the refractive index in the simulation does not match the previously set parameter value. This might be a consequence of an unknown programming error in paraPropPython, that has not been fixed yet. To avoid this error, a scaling of the time axis is carried out in the following way: first, the time axis is normalized to 1 by dividing by the “wrong” TOF at a given range and any depth. Since the correct TOF of a wave propagating through a dielectric medium (i.e. the *direct* transmitted signal in ice) with given refractive index is known, the time axis can be multiplied by this correction factor. Hence, the TOFs of the *reflected* pulse are scaled in the same manner, producing correct results. In order to find this correction factor, the following formula should be considered:

$$t_{corr} = \frac{s}{v} = \frac{r \cdot \sqrt{\epsilon'_{r,medium}}}{c} \quad (5.5)$$

and thus

$$T_{new} = \frac{T_{old}}{TOF_i} \cdot t_{corr} \quad (5.6)$$

where  $T_{old}$ ,  $T_{new}$  are the time axes before and after the correction and  $TOF_i$  is the TOF at any depth  $i$ . The refractive index of the simulation  $n_{simul}$  can be determined by performing a linear fit of the TOFs across the whole range. It is given then by the slope (cf. Fig. 5.13).



**Figure 5.13:** Linear Fit of the TOFs across the receiving antenna range in order to determine the refractive index of simulation  $n_{simul}$ .

## 5.3 Results for the Small-Scale Simulation

In this section, the results of the small-scale simulation for pure ice and on Enceladus are discussed.

## A-Scans

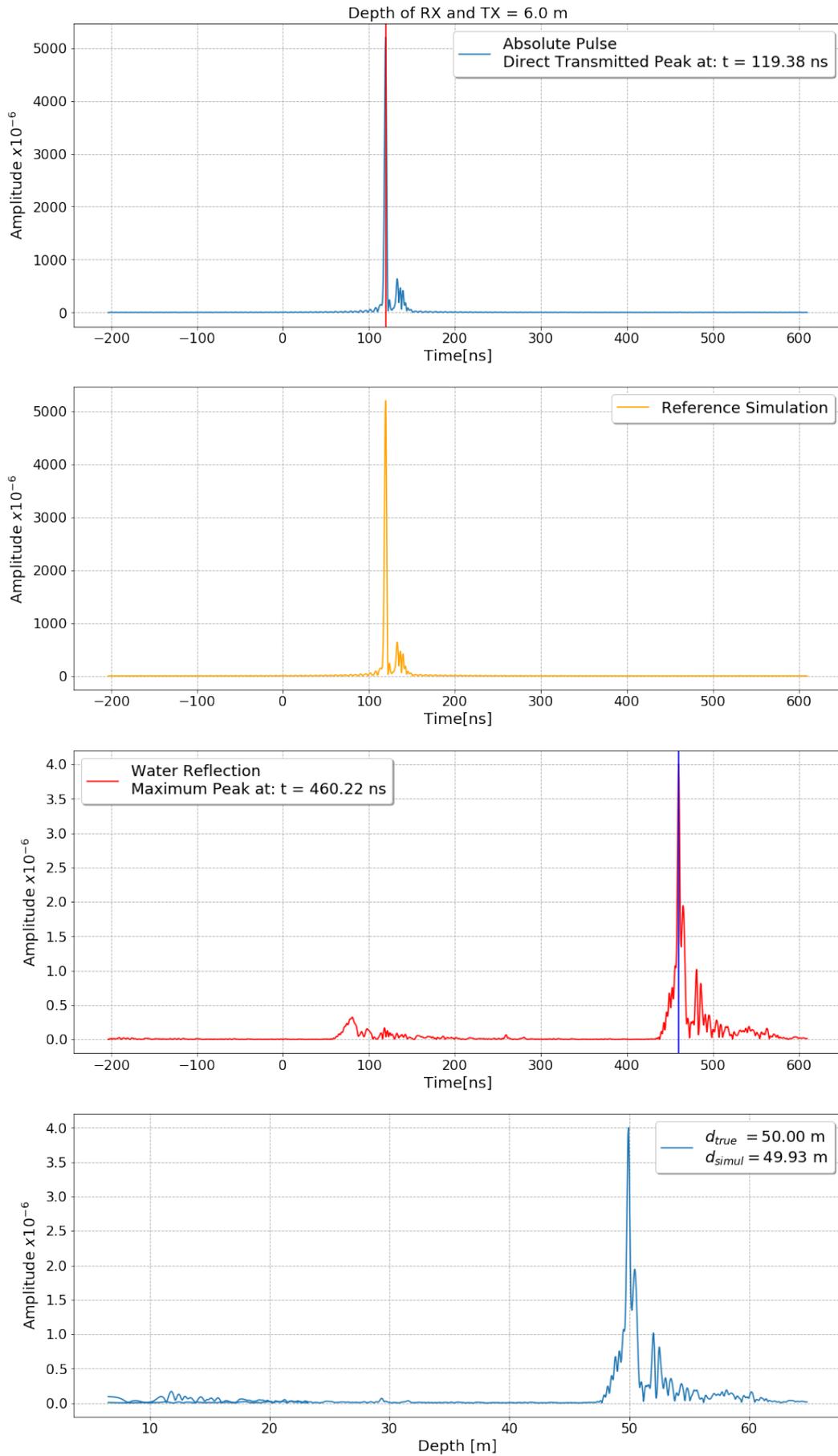
A-Scans were taken for each depth and range in the simulation geometry. In Figure 5.14 the absolute pulse of the directed transmitted peak, the reference simulation, the subtraction of both and a “depth-spectrum” at a transceiver depth of 2 meters and a range of 20 meters are shown for the pure ice scenario. By means of Equation 5.4 the time axis is transformed into a depth axis, showing a broad spectrum of water pocket depths. Both, the TOFs of the direct transmitted pulse and of the maximum peak of the water reflection are given in the legends. Note the amplitude in Figure 5.14 is scaled by  $10^{-6}$ . Note also the decrease in the amplitude of the water reflection compared to the absolute pulse of the main simulation. The true and calculated water pocket depth are in good agreement with each other with the antennae being at 6 m depth (deviation is less than 1%). Here it was assumed that the first and strongest arriving signal is the reflection according to Figure 5.5 that is required to calculate the water pocket depth. The results (cf. Fig. 5.17) show that the algorithm using a simple geometrical derivation is precise enough to describe the reflection from the flat water pocket and calculates the the water pocket depth to be:

$$d_{simul}^{ice} = 51.46 \text{ m} \quad (5.7)$$

which corresponds to a deviation of 2.84 % from the true value. The reflection spectrum consists of multiple, consecutively arriving reflected signals (cf. Fig. 5.14), which are partly superimposed on each other. They might be consequences of interferences and second-order reflections. The latter means a reflection that scattered first at the ice then at the ice-vacuum boundary layer at the surface and reflects back to the RX. However, the calculation of the water pocket depth is based on the simple assumption that the reflection occurs exactly at  $R/2$ , equivalent to Fermat’s principle of least time. The linear behavior of the water pocket depths increasing with increasing depth of the antennae cannot be conclusively clarified and justified. These observations will be important when discussing the large-scale simulations and possible explanations will therefore be given below.

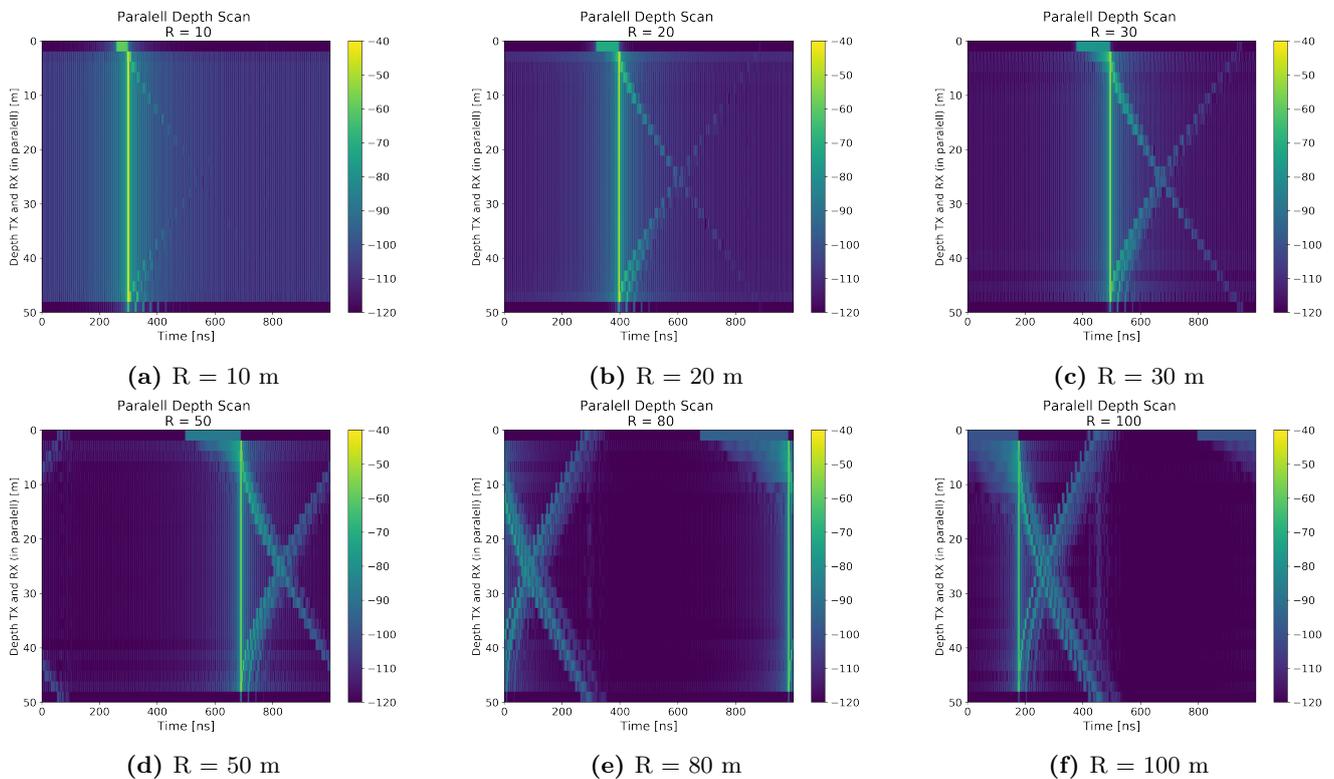
## B-Scans

If the TX and the RX are lowered down in parallel, sampling the pulses across the whole range of the depth profile, a B-Scan can be produced. Figure 5.15(a-f) shows B-Scans at different distances  $R$  between the TX and the RX. A B-Scan displays the amplitude of the electric field along the depth axis and with the corresponding time-of-flight of the signal component. The colorbar is in a logarithmic scale, i.e., the amplitude is given in units of decibels. The vertical yellow line indicates the direct transmitted signal. In the case of a distance of  $R = 10 \text{ m}$  (cf. Fig. 5.15(a)) between the two antennas, this means that a radio wave that was reflected neither at the ice-vacuum boundary layer nor at the ice-water boundary layer took about 300 nanoseconds to get from TX to RX in this simulation. The oblique lines represent the vacuum reflection and the water reflection, respectively. Thereby, the yellow line going from the upper left to the lower right represents the reflection of the vacuum and vice versa, the water reflection is represented by the yellow line going from the lower left to the upper right. All three signals move to the right in the time-domain with increasing range of the antennae, what is to be expected. Since the sample interval is 1000 ns, the yellow features do not exceed this value of time value and start again at 0 ns going



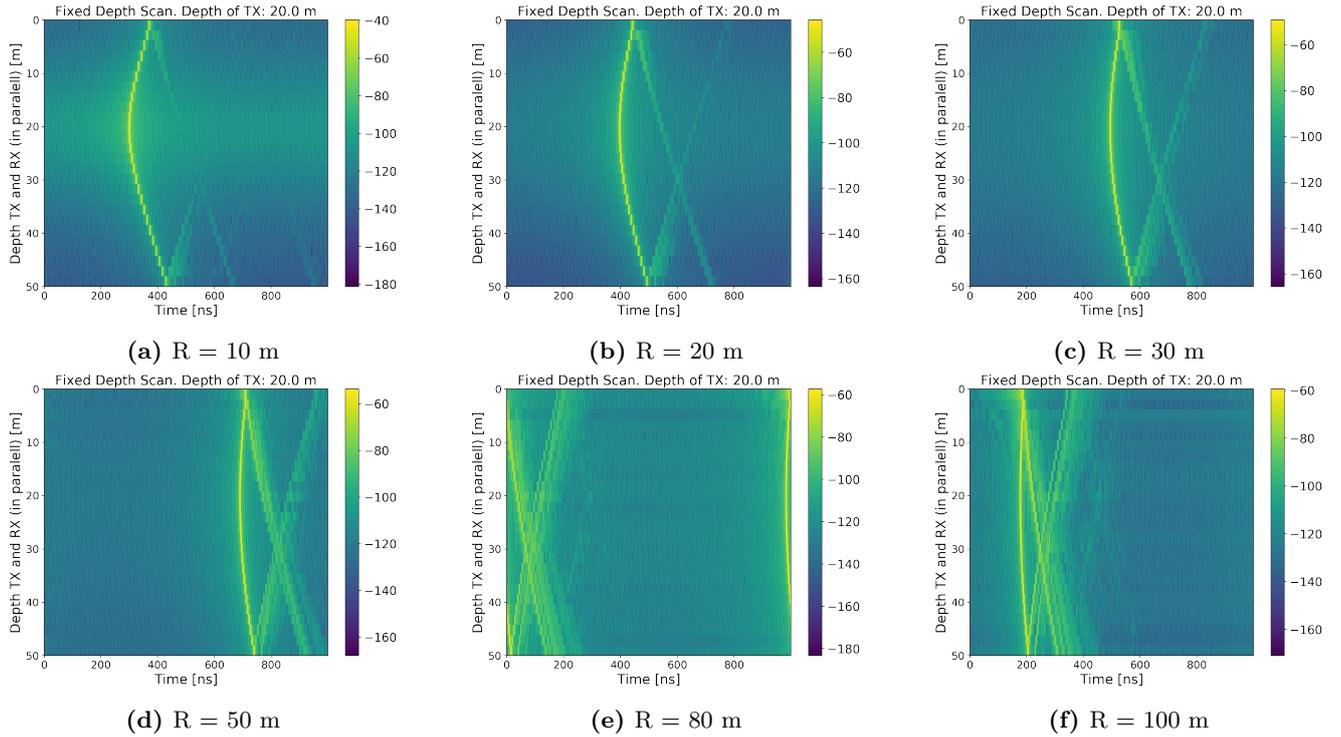
**Figure 5.14:** A-Scans of the unfiltered-, reference-, and filtered signal (blue, yellow, red) and derived water pocket depths (blue) of the small-scale simulation in ice.

from the left to the right. A B-Scan can provide an approximate estimate of the amplitude. Compared with the amplitude at a distance of 10 m of the antennae to each other, which is at -40 dB, the amplitude decreases to roughly -55 dB. In general, the reflections are weaker in amplitude and are more smeared out than the direct transmitted signals since they are incumbent upon scattering and spreading losses. As already seen in the A-Scans, the reflection spectrum consists of multi-component peaks, i.e. reflection could occur in more than one reflection point, or the scattering process show dispersive behavior. This will be further elaborated when analyzing the large-scale simulations. The oblique yellow features of the water reflection show a sort of curvature or kink in vicinity of the ice-water interface. This might have to do with the simulation being imprecise by allowing a strongly varying refractive index  $n = 1 + \delta n$  (cf. chap. 4.2) as the simulation was designed for terrestrial glaciers with density fluctuations. It should also be considered that wide-angle approximation still produces some errors. The simulation can be enhanced by using very-wide-angle approximations such as Padé-methods (Levy and Zaporozhets, 2000). Since the implementation of an algorithm and the explanation of this method would go beyond the scope of this thesis, this can be read in Apaydin and Sevgi (2017).



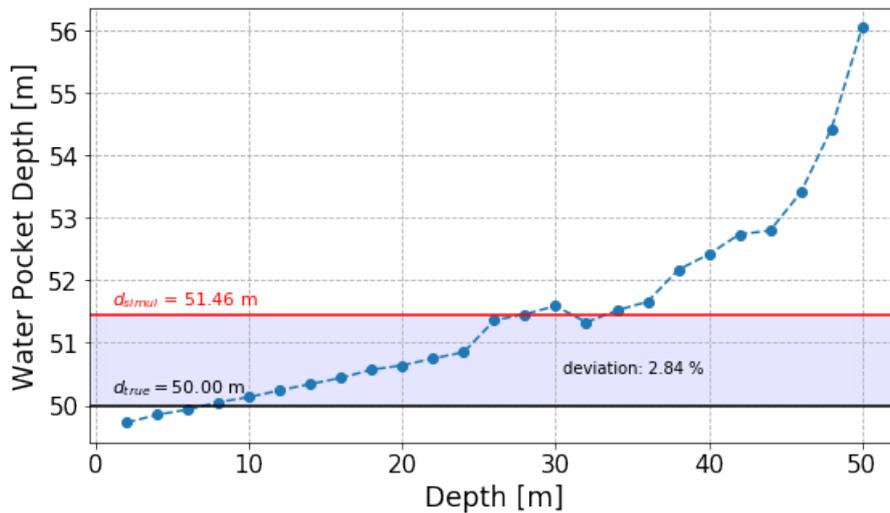
**Figure 5.15:** Crosshole GPR (or Parallel Depth Scans) across the whole range in  $x$ -direction.

A commonly used technique in Borehole GPR applications is a so-called fixed depth scan as it is less effort. The TX is maintained at a fixed depth while the RX is lowered down across the depth range. Figure 5.16(a-f) shows fixed depth scans for several TX-RX pairs at different distances to each other and a fixed depth of 20 m. The direct transmitted signal is now (symmetrically) sickle-shaped as the radio waves take the shortest distance if the TX and the RX are at the same level, while they take longer as they move away from the source depth. For longer distances the radius of curvature of the sickle becomes smaller. Further, for long distances the water reflection becomes ambiguous and smears out.



**Figure 5.16:** Fixed Depth Scans. The TX is at a fixed depth of 20 m and the RX is lowered across the whole depth range in  $z$ -direction. The colorbar has a wider extent as in the parallel depth scan plots.

### Water Pocket Depth



**Figure 5.17:** Calculated water pocket depths in dependence on the transceiver depth. The red line gives the average of the 25 values.

## 5.4 Results for the Large-Scale Simulation

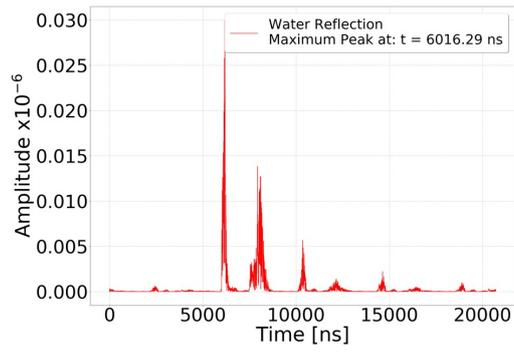
In this section, the results of the large-scale simulation are presented.

## A-Scans

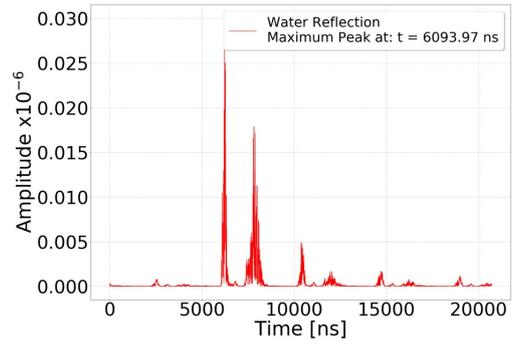
A-Scans from S2, i.e. a 4-layer simulation geometry with a flat water pocket but on a larger scale in contrast to S1, are shown in Fig. 5.18(a-h). A radio wave that directly (without reflections) propagates from TX to RX with a distance of 495 m to each other in a crosshole GPR takes 2953.58 ns in pure ice, 1808.69 ns in the snow and 1651.10 ns in vacuum. The strongest first arriving signal occurs in the snow layer and has an amplitude of  $0.030 \cdot 10^{-6}$  with a TOF of 6016.29 ns (cf. Fig. 5.18a), which is slightly more than three times than the amplitude of the direct transmitted signal. It can be observed in the water reflection spectrum how double peaks occur (cf. Fig. 5.18), moving closer to each other for increasing antenna pair depth until they merge completely into each other at a transceiver depth of 115 m. These double peaks are in turn composed of ambiguous smaller peak components. This behavior could be the cause of non-water reflections, since they have been almost but not completely filtered out by the reference simulation. Another possible explanation is the occurrence of second-order (or more generally: higher-order) reflections, but they cannot be justified with the TOFs because they would have to be higher for the first double peaks at least. Another explanation is that there might be some errors in the simulation. In this specific case this would mean: the stronger peak of the double peaks is the reflection according to the abovementioned single-layer scheme where the reflection occurs at  $R/2$ . The other peak could be a reflection of a radio wave that does not exactly scatter in the geometric center between TX and RX, but in a small environment around this point. This could be the consequence of the simulation allowing strongly varying refractive indices as well as it was shown in the parallel depth scans in the small-scale simulation, affecting the propagation direction of the wave. Initially, the code was designed for such cases (a boundary layer between ice and vacuum), but the change in refractive index is more significant for the case of ice and saline water. Since the distance between the double peaks decrease with increasing depth of the antennae, this error disappears with decreasing distance between antennae and water pocket (reflection target). The PE models the energy propagation within a paraxial cone in a certain angular range. Thus, it is most precise for the horizontal direction and less precise if the reflection occurs vertically. As the antennae are lowered down, the incident angle becomes smaller and the the vertical reflection component is lower. A further possible explanation is the occurrence of dispersive effects that were hinted when discussing the small-scale simulation: a radio wave incident on a conductive optical medium like saline water could be split into multiple frequency constituents at the boundary layer, generating many smaller waves whose direction of propagation depends on the respective frequency. Interactions like interferences between these waves are conceivable as well. However, it cannot be conclusively clarified within the frame of this work and must be investigated by modification of the simulation code.

## B-Scans

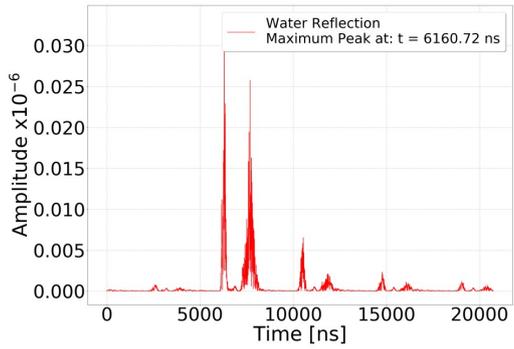
A parallel depth scan of S4 with a distance of  $R = 550$  m between TX and RX is shown in Figure 5.19. The yellow vertical feature, corresponding to the direct transmitted signal and the yellow oblique feature, corresponding to the surface reflection at the sintered snow layer are clearly recognizable. The other oblique line in the opposite direction, that appeared in S1, does not appear here, as the water pocket is triangular and not flat. From this B-Scan, the triangular water pocket cannot be identified. The radio



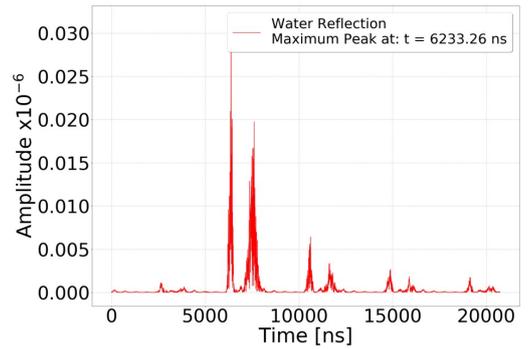
(a) transceiver depth: -25 m (snow-vacuum boundary layer)



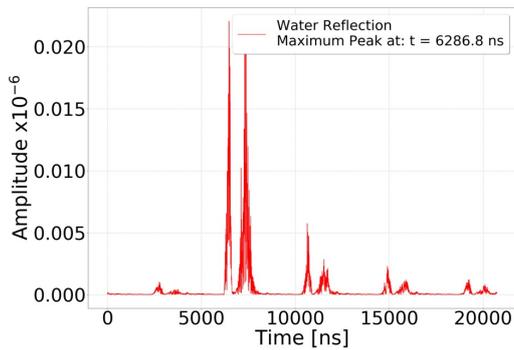
(b) transceiver depth: -5 m (snow)



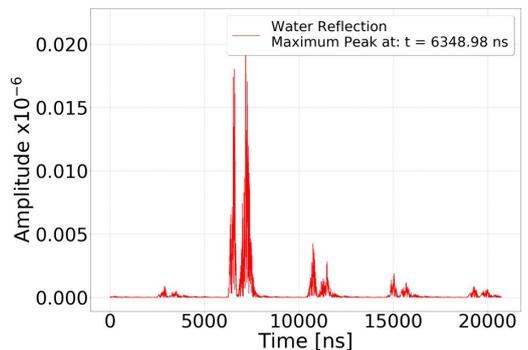
(c) transceiver depth: 15 m (pure ice)



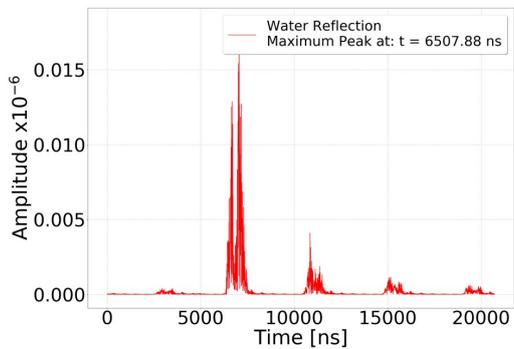
(d) transceiver depth: 35 m (pure ice)



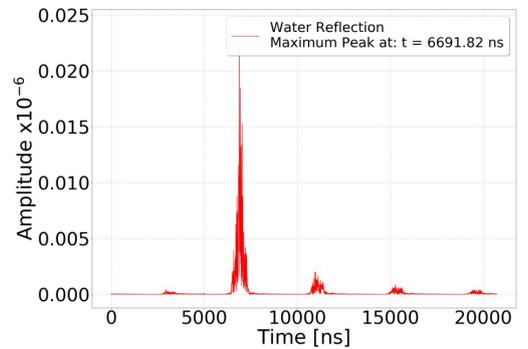
(e) transceiver depth: 55 m (pure ice)



(f) transceiver depth: 75 m (pure ice)



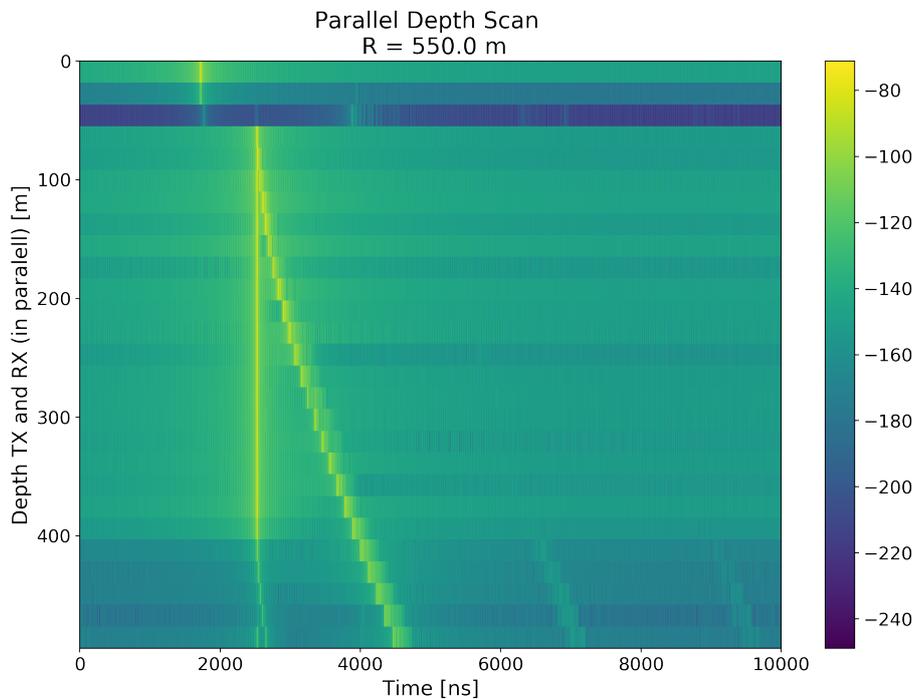
(g) transceiver depth: 95 m (pure ice)



(h) transceiver depth: 115 m (pure ice)

**Figure 5.18:** A-Scans of S2 showing the water reflection spectrum in the time domain at a distance of  $R = 495$  m.

waves might be reflected in several distinct directions as the shape of the water target is irregular. Also, the simulation is designed for flat boundaries, since for split-step/Fourier methods it is difficult to model appropriate boundary conditions (Ehn, 2019) that are therefore not put in by hand, but instead the fields are reflected from the surface as if there were a density fluctuation within a continuous  $n(z)$ -profile (Prohira et al., 2021). On the other hand, by looking at the colormap, the amplitude of the direct signal decreases significantly between 400 and 500 m depth where the water pocket is located. Two smaller and weaker oblique lines occur below 400 m depth and around 7000 ns and 9000 ns, respectively. Those features, however, are hard to interpret and cannot yet be explained. The effect of the sintered snow layer in S4 is present: the refraction, that is, a shift in the time domain of the direct pulse in all 4 layers distinguishable by different green-blue tones, and most clearly at the sintered snow-pure ice interface is visible. As the imaginary parts of the complex permittivity of sintered snow and the conductive granular snow were set to be equal (cf. Tab. 5.2), there is no time shift at their interface. However, an attenuation effect of the sintered snow can be seen from the amplitude, in accordance with the transmission and reflection coefficients (cf. Tab. 5.3).

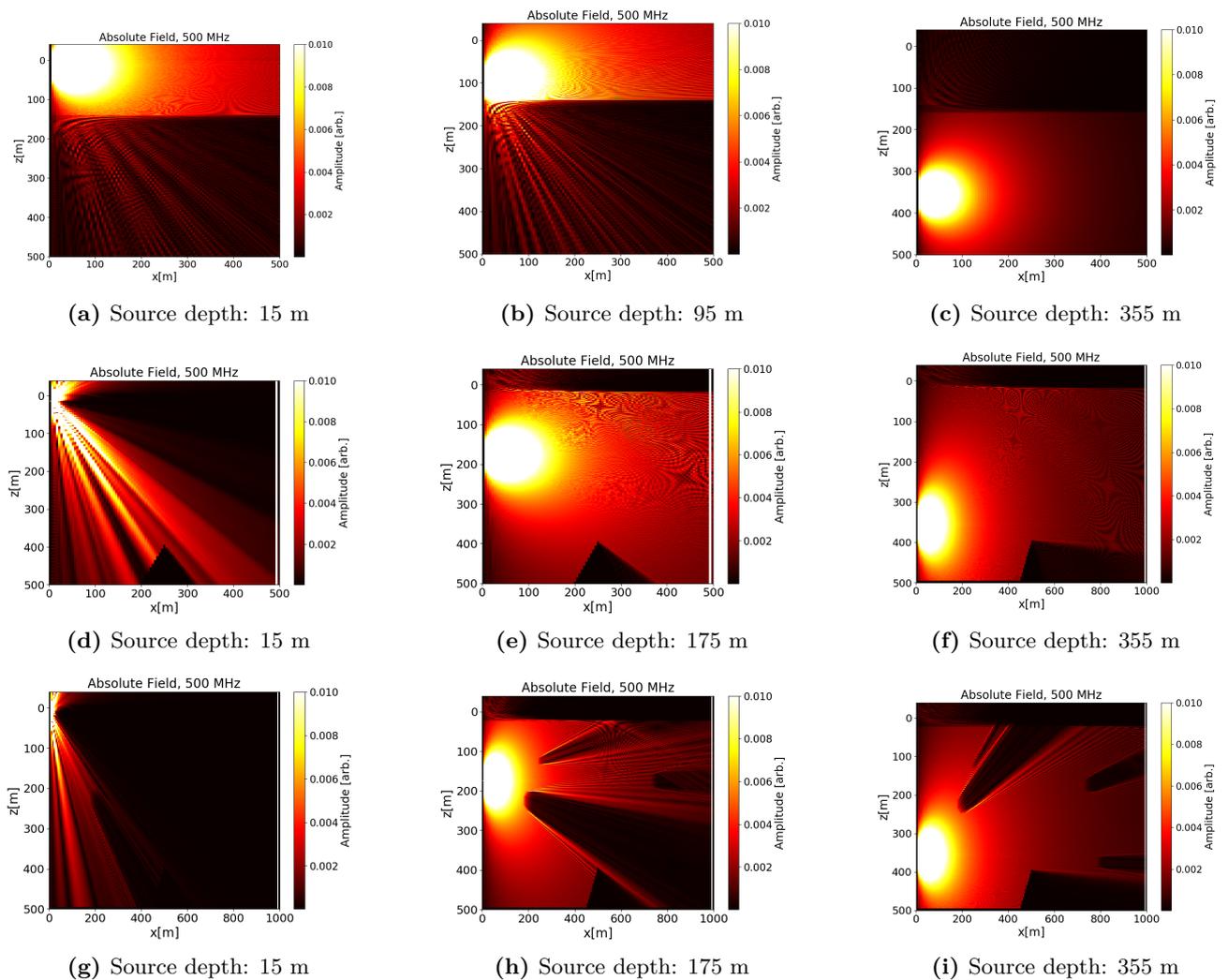


**Figure 5.19:** Crosshole GPR with ZOP (or Parallel Depth Scan) in the large-scale simulation S3 based on the 4-layer structure with a triangular water pocket on Enceladus.

## Absolute Field

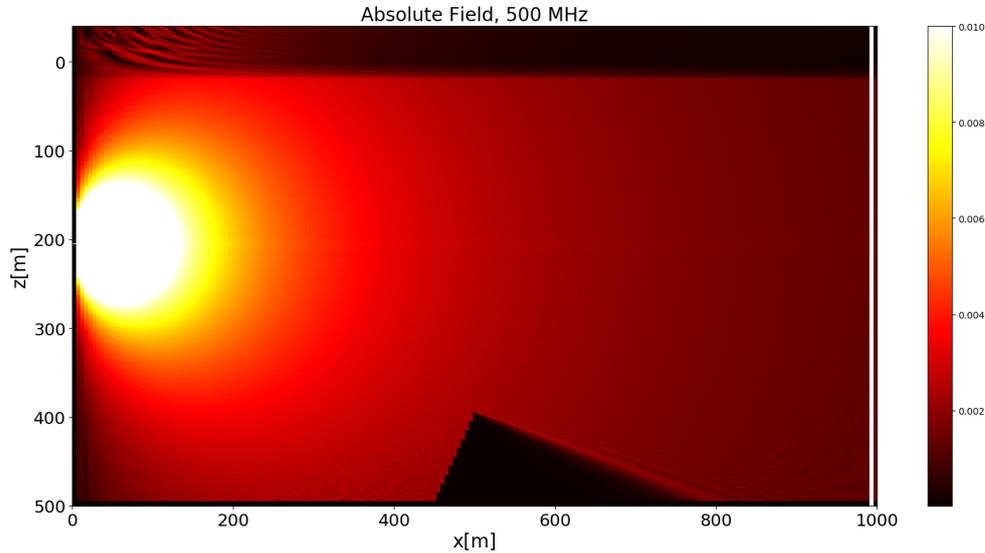
The absolute field plots of S1, S3 and S5 are shown in Figure 5.20(a-i). If the source is buried in the conductive snow layer, a large part of the signal is attenuated in a triangle-shaped area (this does not refer to the water pocket) that is horizontal to it (cf. Fig. 5.20d). If the geyser-deposited snow layer on Enceladus has about the same impurity level as assumed here, this would mean that measurements according to the Vertical Radar Profile method, where the receiver antennas are on the surface, could

not adequately reproduce large parts of the subsurface environment, due to high attenuation of the snow layer. Hence, borehole methods, where the antennae are located beneath the snow layer, seem to be more suitable for such a scenario. However, the triangular shape of the water pockets can be clearly reproduced in those simulations. If the source is lowered down to higher depths, then the triangle, which was set to be isosceles, becomes non-isosceles (cf. Fig. 5.20e) or even loses its triangular shape due to its own shadow (cf. Fig. 5.20f). Small parts at the rear edge of the triangle are nevertheless illuminated. Here the wave character of the radio source becomes apparent and indicates a correct representation on the part of the PE simulation methodology. A melting probe must avoid suddenly appearing obstacles like meteoric rocks. They can be seen if the antennae are embedded in the ice (cf. Fig. 5.20h & 5.20i) but not when they are embedded in the snow (cf. 5.20g). They lose their circular shape for the same reasons as for the triangle and even prevent to identify the water pocket.

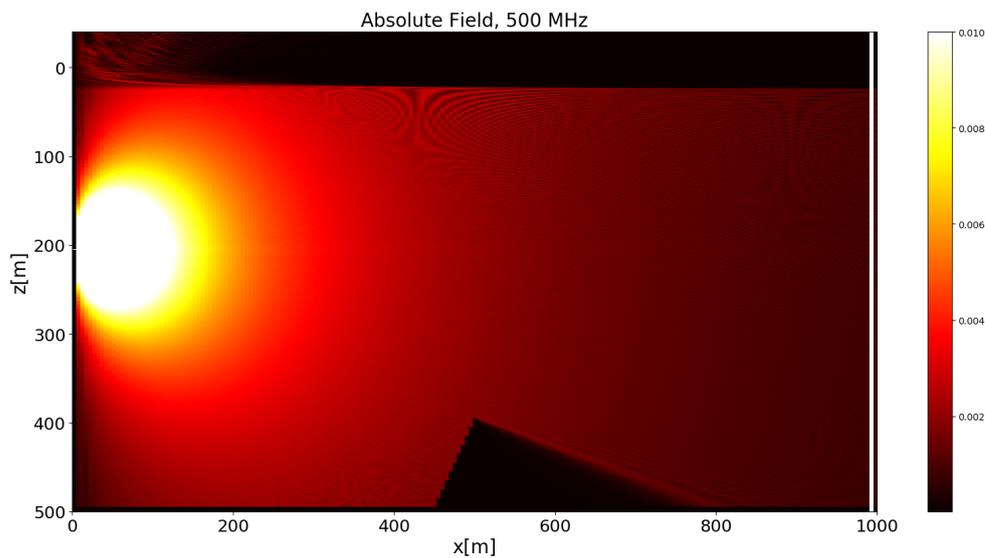


**Figure 5.20:** Absolute Field of S1 (a-c), S3 (d-f) and S5 (g-i) with a single transmitter at different source depths, no receivers and run at a single frequency of  $f = 500$  MHz. The amplitude is in arbitrary units. The wave-like character is noticeable from the lighter and darker areas in the pattern especially near the boundary layers where the refraction and reflection occur. The triangular water pocket, which absorbs the waves to a great extent, is displayed differently by the diffraction: it casts shadows differently depending on the source depth. Similar arguments from the optics are made with the circular meteoric rocks. The water pocket is shifted to 150 m in (a-c).

The absolute field at  $f = 500$  MHz with a single transmitter and no receivers of S3 and S4 are shown in Figure 5.21 & 5.22. The additional sintered snow layer that due to its lower porosity has a higher conductivity and attenuation constant, has a clear effect on the absorption at the boundary layers.



**Figure 5.21:** Absolute field of S3 at a source depth of 200 m. Above the pure ice layer there is a layer of unconsolidated geyser deposits (conductive snow) that absorbs and attenuates the radio waves.



**Figure 5.22:** Absolute field of S4 at a source depth of 200 m. Above the pure water ice layer there is a layer of sintered geyser deposits (sintered snow) and a layer of unconsolidated geyser deposits (conductive snow). Both surface ice layers have a thickness of 12.5 m. In contrast to Fig. 5.21 the radio waves are more absorbed due to the higher conductivity of the sintered snow layer, which can be seen in the top left corner region of the color map.

## 5.5 Discussion

The previous results show that borehole ground penetrating radar methods provide numerous opportunities in order to describe radio wave propagation in a subsurface environment of a body with a multiple-layer simulation geometry.

### *Benefits*

- A-Scans reproduce reflection spectra correctly in the time domain
- B-Scans enable to distinguish different reflections from each other and can assign them to the corresponding scattering layer
- With the aid of simplistic geometry considerations the depth of a flat scattering target could have been derived within a 5 % error limit.

### *Problems and Enhancements*

- For more complex targets the assumptions made here are not sufficient and more elaborated mathematical and physical considerations like boundary conditions have to be incorporated
- The crosshole GPR B-Scan was not able to show the triangular water pocket. The code must be extended with regard to non-flat boundary layers
- The code has errors regarding refractive index and time axis
- The PE-method applied here allows slightly varying refractive indices in an otherwise homogeneous medium. These has an impact on the reflection signals

## Conclusion and Outlook

This thesis aimed to apply a new simulation method, the Parabolic Equation, an approximation of Maxwell's equations solution, in order to describe radio wave propagation through different homogeneous dielectric media with multiple boundary layers and scattering problems. To that end, based on sufficiently precise assumptions, chemical models and measurement data from Cassini space probe, approximately realistic dielectric properties of the surface and subsurface ice of the Saturnian moon Enceladus like permittivity, conductivity and attenuation could have been derived. In comparison with FDTD-methods, which solve Maxwell's equations directly without any approximations, but are more commonly used, PE-methods are computationally more efficient, even if not being as accurate in the nano-scale.

Within the scope of this work, application purposes comprise the calculation of distances to obstacles like circular meteoric rocks that a melting probe would encounter when drilling through the ice during a space mission to Enceladus, and a water-bearing crevasse, assumed to be flat or triangular here, in differently complex simulation domains. The main tool is to look at the time-of-flight (TOF) of both the direct transmitted signals and the signals reflected from the boundary layers. Concerning this, the methodology includes creating Gaussian pulses emitted by transmitting antennas (TX), which are decomposed in its frequency constituents by Fourier transform and then recomposed in their original states again after being sampled at receiving antennas (RX). After that, so-called *A-Scans* that represent the received pulse in the time-domain at each TX depth and RX range, and *B-Scans* that combine multiple A-Scans across the whole depth range, can be created. This works according to the principle of the Borehole Ground-Penetrating Radar, where TX and RX are lowered down in parallel, that is, at the same depth. The distance calculation was successfully performed. By using a single-layer geometry, the simulation provides a depth that deviates only 2.48 % from the true value. However, this research raises the question of multiple-layer scattering and irregular shapes of objects embedded in the ice, so that statements with regard to the shape and the location of the objects could only have been made qualitatively instead of quantitatively. The A-Scans reveal important information about attenuation and higher-order reflections, while the B-Scans disclose the inaccuracy of the PE-method designed for domains with slightly and strongly changing refractive index such as terrestrial glaciers.

Based on these conclusions, future research should consider the polarization of the electric field and boundary conditions of electromagnetic waves incident on more complex geometries. The simulations were made assuming a flat surface geometry and flat boundary layers. More accurate simulations would include an uneven surface using topography data of Enceladus and require an update of the code with regard to the PE (for example see the chapter *PE and Terrain Modeling* in Apaydin and Sevgi (2017)). Due to the overestimation of the TOF, the code should be corrected with regard to this error.



## Simulation Results

**Table A.1:** Calculated water pocket depths in a range of 2 to 50 m depth in increments of 2 m.

$d$ [m]	$d_{WaterPocket}$ [m]
2	49.72
4	49.85
6	49.93
8	50.05
10	50.13
12	50.24
14	50.34
16	50.44
18	50.57
20	50.64
22	50.75
24	50.85
26	51.36
28	51.46
30	51.59
32	51.33
34	51.52
36	51.66
38	52.17
40	52.42
42	52.74
44	52.80
46	53.41
48	54.43
50	56.05

# B

## Conductivity of Enceladean Surface Ice and Subsurface Salt Water Ocean

It is likely that a subglacial, salty ocean also exists on Jupiter's moon Europa. Moore (2000) proposes models of radar absorption in European ice that could be purposeful in order to detect the ocean by an orbiting radar sounder. The attenuation of the electromagnetic waves depends on the ice-chemistry, for which some geochemical models are given in this paper that take into account temperature variation with depth in a certain frequency range. Walker et al. (2010) present an equivalent approach to the same problem on Enceladus. In the following a rough idea of Enceladus' ice- and water-chemistry will be discussed whence dielectric properties like refraction, conduction and attenuation will be derived.

### Conductivity of Enceladean Ice

In this section, the main dielectric parameters of Enceladean snow-like granular surface ice will be derived, based on the assumption that it is mainly contaminated with sodium compounds.

The molar concentration (molarity) of ionic sodium ( $N^+$ ) is given by:

$$c = \frac{n_{Na^+}}{V} = \frac{N_{Na^+}}{N_A \cdot V} \quad (\text{B.1})$$

where  $N_A$  is the Avogadro constant and  $V$  is the volume of an ice grain assumed to be spherical in shape:

$$V = \frac{4}{3}\pi R_g^3 \quad (\text{B.2})$$

with a radius of  $R_g \simeq 150 \mu\text{m}$  (cf. 2.2 and Brown et al. (2006)). In order to find the number of sodium particles, first calculate the number of water molecules by their mass

$$m_w = \rho_w \cdot V = \rho_w \cdot \frac{4}{3} \cdot \pi R_g^3 = 920 \frac{\text{kg}}{\text{m}^3} \cdot \frac{4}{3} \pi \cdot (150 \cdot 10^{-6} \text{ m})^3 = 1.300 \cdot 10^{-8} \text{ kg} \quad (\text{B.3})$$

The number of water molecules can be calculated through the molar mass of water:

$$N_{\text{H}_2\text{O}} = \frac{3.083 \cdot 10^{-14} \text{ kg}}{18\text{u}} = 4.353 \cdot 10^{17} \quad (\text{B.4})$$

Since  $\frac{N_{Na^+}}{N_{H_2O}} \gtrsim 10^{-3}$  for sodium-rich Type-III particles that predominantly contain NaCl and Na<sub>2</sub>CO<sub>3</sub> and/or NaHCO<sub>3</sub> (cf. chapter 2.2), it is:

$$N_{Na^+} = 4.353 \cdot 10^{15} \quad (B.5)$$

Insertion yields the following concentration of sodium in the ice grain:

$$c_{Na^+} = \frac{4.353 \cdot 10^{15}}{6.022 \cdot 10^{23} \text{ mol}^{-1} \cdot \frac{4}{3}\pi \cdot (150 \cdot 10^{-6} \text{ m})^3} = 51.14 \frac{\text{mol}}{\text{m}^3} = 0.051 \frac{\text{mol}}{\text{l}} = 5.114 \cdot 10^4 \mu\text{M} \quad (B.6)$$

The electric conductivity  $\Lambda$  of the impure ice can be calculated through the molarity and molar conductivity of sodium ( $\beta_{Na^+} \sim 1.5 \frac{\mu\text{S}}{\mu\text{M}}$ ). It will be looked at the high-frequency limit of  $\sigma$ :

$$\sigma_{\infty}^{\text{Enc, snow}} \simeq \sigma_{\infty, \text{pure ice}} + \beta_{Na^+} \cdot c_{Na^+} + \underbrace{\sum_i \beta_i c_i}_{\text{other impurities, negligible}} \quad (B.7)$$

## Snow

Further, the molar conductivity depends on the volume filling factor  $v$ , which is given by the porosity  $p = 1 - v$  of the ice considered. A first approach states a  $v^3$ -dependency of the molar conductivity. An even better approximation is given by the Looyenga-model:

$$\beta(v) \simeq \beta(1) \cdot v \cdot (0.68 + 0.32v)^2 \quad (B.8)$$

or a general power law

$$\beta(v) = \beta(1) \cdot (v - v_c)^t \quad (B.9)$$

where  $v_c$  equals to a threshold volume fraction below no conduction can take place and  $t$  is dependent on the lattice structure of the impure ice and is often found to be between 2.5 and 3 but can merely be determined by experimental models (not applicable here). The porosity of Enceladean surface ice is known to be  $p = 0.9 - 0.95 \iff v = 0.05 - 0.1$  (Buratti et al., 2014). Finally, the following equation for the conductivity of sodium-doped Enceladean ice holds:

$$\sigma_{\infty} \simeq \sigma_{\infty, \text{Looyenga}} = \underbrace{\sigma_{\infty \text{ pure}}(1)v(0.68 + 0.32v)^2}_{=0.456 \mu\text{Sm}^{-1}} + \underbrace{\beta_{Na^+}(1)v(0.68 + 0.32v)^2}_{=0.076 \frac{\mu\text{Sm}^{-1}}{\mu\text{M}}} \cdot \underbrace{[Na^+]_{v=0.1}}_{=5.114 \cdot 10^4 \mu\text{M}} \quad (B.10)$$

Insertion of all parameters yields:

$$\sigma_{\infty}^{\text{Enc, snow}} = 3889.127 \mu\text{Sm}^{-1} = 3.889 \cdot 10^3 \mu\text{Sm}^{-1} \quad (B.11)$$

The attenuation constant is given by the following expression (Evans, 1965):

$$\alpha = 0.129 \sqrt{\varepsilon_r} f \cdot \left[ \sqrt{(1 + \tan^2 \delta) - 1} \right]^{-\frac{1}{2}} \quad (B.12)$$

respectively, if  $\tan \delta \ll 1$  as it is for low loss materials such as most ice at radar sounding frequencies (Moore, 2000), then the attenuation constant can be rewritten as:

$$\alpha = 0.0009 \cdot \sigma \frac{\text{dB}}{\text{m}} \quad (\text{B.13})$$

where  $\sigma$  is in  $\mu\text{Sm}^{-1}$ . The attenuation caused by sodium-bearing ice is therefore given by;

$$\alpha_{\text{Na}^+} = 9 \cdot 10^{-4} \cdot 3889.127 \text{ Sm}^{-1} = 3.500 \frac{\text{dB}}{\text{m}} \quad (\text{B.14})$$

The imaginary part of the dielectric constant can be calculated by the electric conductivity:

$$\varepsilon''(\omega) = \frac{\sigma}{\varepsilon_0 \omega} = \frac{\sigma}{2\pi \varepsilon_0 f} = \frac{3889.127 \cdot 10^{-6} \text{ Sm}^{-1}}{2\pi \cdot 8.85 \cdot 10^{-12} \frac{\text{As}}{\text{Vm}} \cdot 500 \text{ MHz}} = 0.139 \quad (\text{B.15})$$

Analogously, one can calculate the conductivity contribution of sodium-poor Type-I particles. Since  $\frac{\text{Na}^+}{\text{H}_2\text{O}} = 10^{-8} - 10^{-5}$  (Postberg et al., 2009), a lower and an upper bound can be specified for the number of sodium particles:

$$1.032 \cdot 10^4 < N_{\text{Na}^+}^{\text{Type-I}} < 1.032 \cdot 10^7 \quad (\text{B.16})$$

and then for the molar concentration:

$$0.511 \mu\text{M} < c_{\text{Type-I}}^{\text{Na}^+} < 511 \mu\text{M} \quad (\text{B.17})$$

Finally, it can be seen that the HF-conductivity contribution of Type-I particles, which is given by

$$\sigma_{\infty}^{\text{Type-I}} = \begin{cases} 0.456 \mu\text{Sm}^{-1}, & \text{if } \frac{\text{Na}^+}{\text{H}_2\text{O}} = 10^{-8} \\ 0.493 \mu\text{Sm}^{-1}, & \text{if } \frac{\text{Na}^+}{\text{H}_2\text{O}} = 10^{-5} \end{cases} \quad (\text{B.18})$$

approaches that of pure ice for the lower bound and is only minimally larger for the upper estimation of  $N_{\text{Na}^+}$ . It will therefore be neglected and for that reason, in the following it will be assumed that near a geyser-active fracture of a Tiger Stripe there will exclusively sediment sodium-rich Type-III particles back to surface, stratifying it in a conductive snow-like layer.

## Sintered Snow

Sintered snow on Enceladus probably has a volume filling factor of  $v = 0.4$ . This corresponds to a porosity of 60 %. Therefore, the conductivity and the attenuation slightly change:

$$\sigma_{\infty}^{\text{Enc, sint.}} = 20034.829 \mu\text{Sm}^{-1} = 20.035 \cdot 10^3 \mu\text{Sm}^{-1} \quad (\text{B.19})$$

$$\alpha = 18.031 \frac{\text{dB}}{\text{m}} \quad (\text{B.20})$$

and the imaginary part of the permittivity at  $f = 500 \text{ MHz}$  is:

$$\varepsilon'' = 0.721 \quad (\text{B.21})$$

## Conductivity of Enceladus' Salt Water Ocean

A transmitted signal faces more attenuation as the conductivity, which changes with physical parameters like salinity, increases. Sea water on Earth has a high conductivity ( $\sim 4 \text{ Sm}^{-1}$ ) in contrast with pure water ( $\sim 0.005 - 0.01 \text{ Sm}^{-1}$ ) (Karagianni, 2015). For all Debye parameters in the Debye model (cf. Eq. 3.72) linear interpolation functions can be deduced:

$$\varepsilon_s(T, S) = a_1(T) + S \cdot a_2(T) \quad (\text{B.22a})$$

$$\tau(T, S) = b_1(T) + S \cdot b_2(T) \quad (\text{B.22b})$$

$$\sigma(T, S) = c_1(T) + S \cdot c_2(T) \quad (\text{B.22c})$$

$$\varepsilon_\infty(T) = 6.4587 + \mathcal{O}(T) \quad (\text{B.22d})$$

where  $a_1$ ,  $a_2$ ,  $b_1$ ,  $b_2$ ,  $c_1$  and  $c_2$  are polynomial functions up to the 6<sup>th</sup> order and can be looked up in Ellison et al. (1998). Further, Matson et al. (2012) estimates a relatively warm ocean temperature of  $\sim 0^\circ\text{C}$ , two degrees above the freezing temperature of saline water. Insertion of this temperature and a lower and an upper bound of the salinity yields the following parameters:

$$\sigma(T = 0^\circ\text{C}, S = 10/30 \text{ ppt}) = \begin{cases} 0.861 \text{ Sm}^{-1}, & \text{if } S = 10 \text{ ppt} \\ 2.409 \text{ Sm}^{-1}, & \text{if } S = 30 \text{ ppt} \end{cases} \quad (\text{B.23})$$

All terms except for the constants can be neglected since  $T = 0^\circ\text{C}$ . The relaxation time is

$$\tau(T = 0^\circ\text{C}, S = 10/30 \text{ ppt}) = \begin{cases} 17.303 + 10 \frac{\text{g}}{\text{kg}} \cdot (-6.72 \cdot 10^{-3}) = 17.240 \text{ [ps]}, & \text{if } S = 10 \text{ ppt} \\ 17.303 + 30 \frac{\text{g}}{\text{kg}} \cdot (-6.72 \cdot 10^{-3}) = 17.101 \text{ [ps]}, & \text{if } S = 30 \text{ ppt} \end{cases} \quad (\text{B.24})$$

and is lower with respect to the relaxation time of fresh water, i.e. the doped water molecules respond more quickly to an external field. The static permittivity holds:

$$\varepsilon_s(T = 0^\circ\text{C}, S = 10/30 \text{ ppt}) = \begin{cases} 81.820 + 10 \frac{\text{g}}{\text{kg}} \cdot 0.12544 = 83.074, & \text{if } S = 10 \text{ ppt} \\ 81.820 + 30 \frac{\text{g}}{\text{kg}} \cdot 0.12544 = 85.583, & \text{if } S = 30 \text{ ppt} \end{cases} \quad (\text{B.25})$$

and the high-frequency value which is independent on salinity is  $\varepsilon_\infty = 6.4587$ . Insertion of all values in Equation 3.72 yields for the real and imaginary part of the complex permittivity:

$$\varepsilon'(f = 500 \text{ MHz}, T = 0^\circ\text{C}, S = 10/30 \text{ ppt}) = \begin{cases} 82.850, & \text{if } S = 10 \text{ ppt} \\ 85.355, & \text{if } S = 30 \text{ ppt} \end{cases} \quad (\text{B.26})$$

$$\varepsilon''(f = 500 \text{ MHz}, T = 0^\circ\text{C}, S = 10/30 \text{ ppt}) = \begin{cases} 35.105, & \text{if } S = 10 \text{ ppt} \\ 90.884, & \text{if } S = 30 \text{ ppt} \end{cases} \quad (\text{B.27})$$

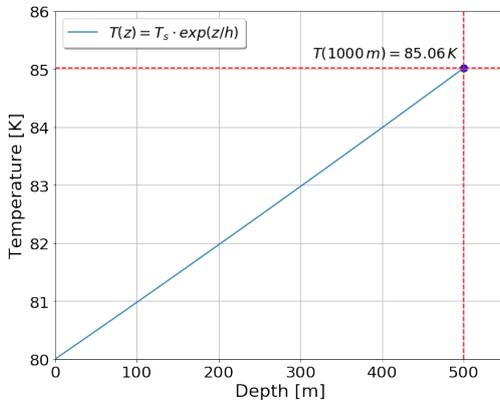
## Temperature Profile of the Ice Crust on Enceladus

It has been shown that the temperature-dependency of the conductivity can be expressed as a series of Arrhenius-functions dependent on the conduction contribution  $C_i$ , activation energy  $E_i$  and a reference temperature  $T_r = 251$  K (Moore, 2000)

$$\sigma(T) = \sum_i C_i \exp\left[\frac{E_i}{k_B} \cdot \left(\frac{1}{T} - \frac{1}{T_r}\right)\right] \quad (\text{B.28})$$

and a model for the temperature profile as a function of depth from the surface for icy moons like Europa and Enceladus was given by Chyba et al. (1998). According to this model, it is:

$$T(z) = T_s \cdot \exp\left(\frac{z}{h}\right) \quad (\text{B.29})$$



where  $T_s$  is the surface temperature at  $z = 0$  and  $h = \frac{b}{\ln T_b/T_s}$  where  $b$  is the ice thickness and  $T_b$  is the temperature at the ice base. The surface temperature at the SPT is roughly 80 K and the temperature at the ice-ocean interface is approximately 270 K. The ice thickness is assumed to be  $b = 10$  km. The simulation geometry will be within a range of 0 up to 500 m depth and therefore:

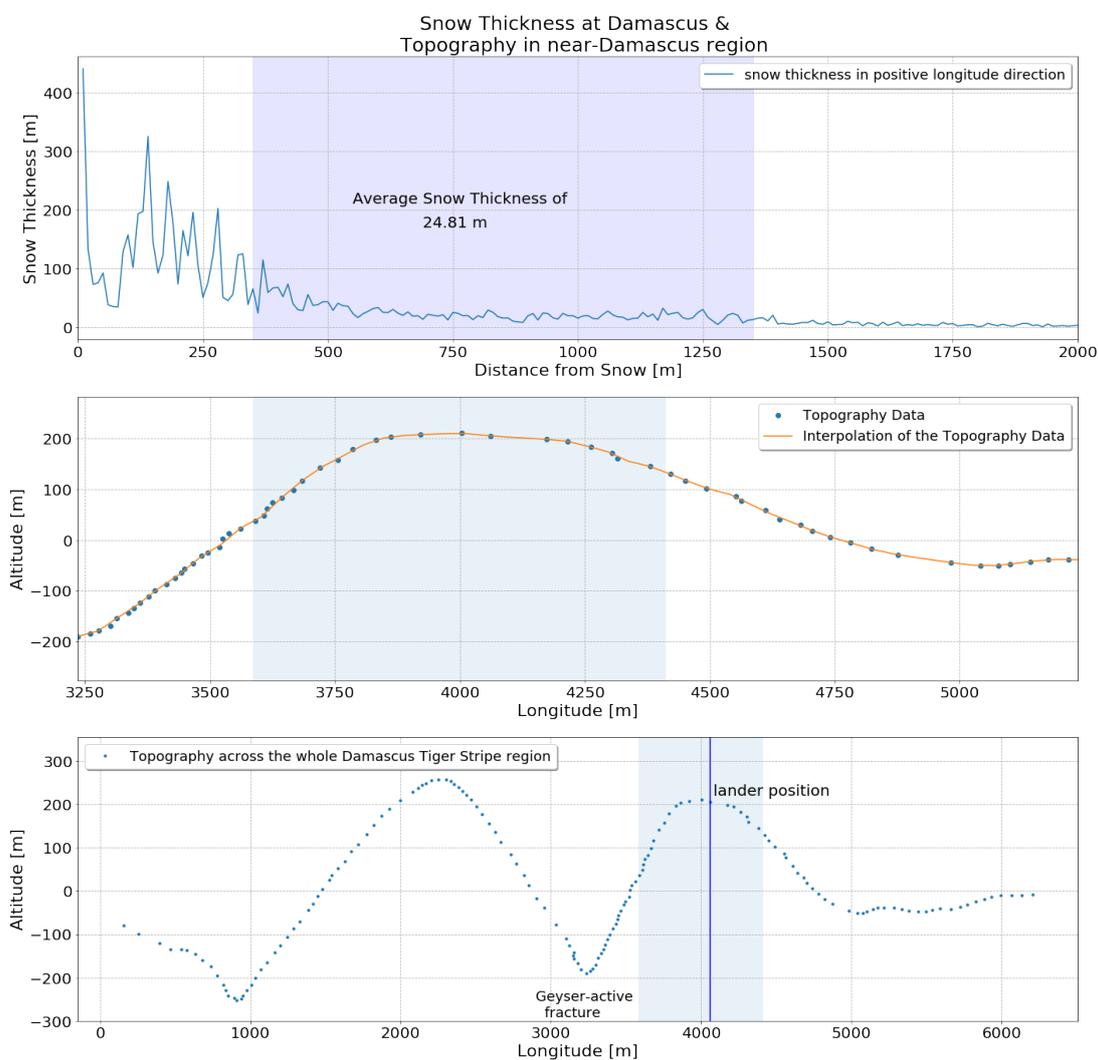
$$T(500 \text{ m}) = 85.06 \text{ K} \quad (\text{B.30})$$

**Figure B.1:** Temperature profile model of Enceladus' ice crust

which is an increase of roughly 5.1 K (cf. Fig. B.1) and will not be considered to be significant.

## Simulation Environment

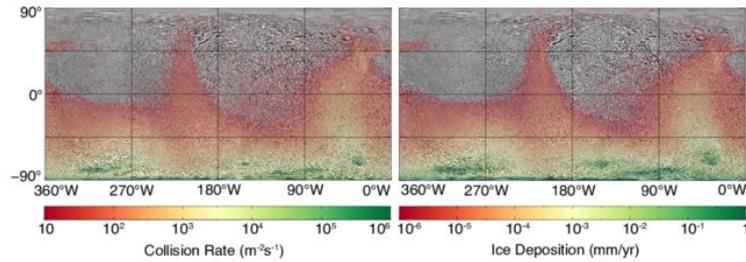
Landing directly at a cryovolcano might be too risky. For an optimal landing position, a certain safety distance and a relatively flat topographical environment must therefore be ensured, and the thickness of the snow layer must also be appropriately chosen.



**Figure C.1:** Snow thickness and 2D topography map of the Damascus Sulcus Tiger Stripe. Sources: Friend and Kyriacou (2021) & Abramov et al. (2015)

In Figure C.1 the snow thickness and a two-dimensional topography map of the Damascus Sulcus Tiger

Stripe region<sup>1</sup> is shown. In order to find an adequate lander position, an interpolation through the topography data was performed and afterwards the flattest environment was found by looking at the minimum of the gradient with `np.gradient` from the `numpy`-library. The result was found at roughly 825 m away from the fracture (blue vertical line in Fig. C.1). The snow thickness in this environment, which extends over 1000 metres as in the simulation, is 24.81 m on average. On this basis, a snow thickness of 25 m is selected in the simulation. In the simulation geometries the snow thickness is maintained *constant* at this value. The snow thickness data Friend and Kyriacou (2021) is based on a model of the geyser particle deposition rate data (cf. Fig. C.2)



**Figure C.2:** Cumulative plume particle deposition on Enceladus' surface in mm/year for the eight sources proposed in Spitale and Porco (2007), particle sizes 0.6 to 15  $\mu\text{m}$  (Southworth et al., 2018).

<sup>1</sup>[https://drive.google.com/drive/u/0/folders/1tqnIHeWyr8C-E\\_-bc1RyTfmoJqCo3fBg](https://drive.google.com/drive/u/0/folders/1tqnIHeWyr8C-E_-bc1RyTfmoJqCo3fBg), Last retrieved: 09-20-2021

# D

## Codes

### paraPropPython.py

```
1 # paraPropPython
2 # c. sbrocco, s. prohira
3
4 import util
5 import numpy as np
6 import time
7 import datetime
8 from permittivity import *
9
10 class receiver:
11     """
12     Parameters
13     -----
14     x : float
15         x position (m)
16     z : float
17         z position (m)
18     """
19     def __init__(self, x, z):
20         self.x = x
21         self.z = z
22
23     def setup(self, freq, dt):
24         """
25         further setup of receiver using simulation parameters
26
27         Parameters
28         -----
29         freq : float array
30             frequencies (GHz)
31         dt : float
32             time step (ns)
33         """
34         self.freq = freq
35         self.spectrum = np.zeros(len(freq), dtype='complex')
36         self.time = np.arange(0, dt*len(freq), dt)
37
38     def add_spectrum_component(self, f, A):
39         """
```

```

40     adds the contribution of a frequency to the received signal spectrum
41
42     Parameters
43     -----
44     f : float
45         corresponding frequency (GHz)
46     A : complex float
47         complex amplitude of received signal (V/m???)
48     """
49     i = util.findNearest(self.freq, f)
50     self.spectrum[i] = A
51
52     def get_spectrum(self):
53         """
54         gets received signal spectrum
55
56         Returns
57         -----
58         1-d complex float array
59         """
60         return self.spectrum[:int(len(self.freq)/2)]
61
62     def get_signal(self):
63         """
64         gets received signal
65
66         Returns
67         -----
68         1-d complex float array
69         """
70         return np.flip(util.doIFFT(self.spectrum))
71
72     def get_frequency(self):
73         """
74         gets frequency array
75
76         Returns
77         -----
78         1-d float array
79         """
80         return abs(self.freq)[:int(len(self.freq)/2)]
81
82     def get_time(self):
83         """
84         gets time array
85
86         Returns
87         -----
88         1-d float array
89         """
90         return self.time
91

```

```

92
93
94 class paraProp:
95     """
96     Parameters
97     -----
98     iceLength : float
99         length of the simulation (m)
100    iceDepth : float
101        depth of the ice simulated (m)
102    dx : float
103        grid spacing in the x direction (m)
104    dz : float
105        grid spacing in the z direction (m)
106    airHeight : float
107        amount of air to be simulated above ice (m). Initialized to 25 m
108    filterDepth : float
109        size of the filtered reason above and below simulated region (m). Initialized
110    to 100 m
111    refDepth : float
112        reference depth for simulation (m). Initialized to 1 m below surface
113    """
114    def __init__(self, iceLength, iceDepth, dx, dz, airHeight=25, filterDepth=100,
115    refDepth=1):
116        ### spatial parameters ###
117        # x #
118        self.x = np.arange(0, iceLength+dx, dx)
119        self.xNum = len(self.x)
120        self.dx = dx
121
122        # z #
123        self.iceLength = iceLength
124        self.refDepth = refDepth
125        self.iceDepth = iceDepth
126        self.airHeight = airHeight
127        self.z = np.arange(-airHeight, iceDepth + dz, dz)
128        self.zFull = np.arange(-(airHeight + filterDepth), iceDepth + filterDepth + dz,
129    dz)
130
131        self.zNum = len(self.z)
132        self.zNumFull = len(self.zFull)
133        self.dz = dz
134        self.refDepth = refDepth
135
136        ### other simulation variables ###
137        # filter information #
138        self.filterDepth = filterDepth
139        self.fNum = int(filterDepth / dz)
140        win = np.blackman(2*self.fNum)
141        filt = np.ones(self.zNumFull)
142        filt[:self.fNum] = win[:self.fNum]
143        filt[-self.fNum:] = win[self.fNum:]
144        self.filt = filt

```

```

141
142     # z wavenumber #
143     self.kz = np.zeros(self.zNumFull)
144     self.kz[:int(self.zNumFull/2)] = np.linspace(0, np.pi/self.dz, int(self.
zNumFull/2))
145     self.kz[-int(self.zNumFull/2):] = np.linspace(-np.pi/self.dz, 0, int(self.
zNumFull/2))
146
147     # index of refraction array #
148     self.n = np.ones(self.zNumFull, dtype='complex')
149
150     #2D Refractive Index Profile - Added by Alex Kyriacou
151     self.n2 = (1 + 1j) * np.zeros((self.xNum, self.zNumFull))
152
153     # source array #
154     self.source = np.zeros(self.zNumFull, dtype='complex')
155
156     # 2d field array #
157     self.field = np.zeros((self.xNum, self.zNum), dtype='complex')
158
159     def get_x(self):
160         """
161         gets x grid of simulation
162
163         Returns
164         -----
165         1-d float array
166         """
167         return self.x
168
169     def get_z(self):
170         """
171         gets z grid of simulation
172
173         Returns
174         -----
175         1-d float array
176         """
177         return self.z
178
179
180     ### ice profile functions ###
181     def set_n(self, method, nVec=None, nFunc=None, nAir=1.0003):
182         """
183         set the index of refraction profile of the simulation
184
185         future implementation plans:
186             - 2-d profiles
187             - complex index of refraction
188
189         Parameters
190         -----

```

```

191     method : string
192         'vector' for vector defined profile
193         'func' for function defined profile
194     nVec : array
195         if method=='vector', defines the index of refraction profile of ice as an
array
196         Precondition: spacing between elements is dz
197         Postcondition: n(z=0) = nVec[0], n(z=dz) = nVec[1], ... , n(z>=len(nVec)*dz
) = nVec[-1]
198     nFunc : function
199         if method=='func', defines the index of refraction profile of ice as a
function
200         Precondition: nFunc is a function of one variable, z, and returns a float
value
201         Postcondition: n(z>=0) = nFunc(z)
202     nAir : float
203         index of refraction of air
204         Postcondition: n(z<0) = nAir
205     """
206     self.n = np.ones(self.zNumFull, dtype='complex')
207
208     ### vector method ###
209     if method == 'vector':
210         nNum = len(nVec)
211         j = 0
212         for i in range(self.zNumFull):
213             if self.zFull[i] >= 0:
214                 if j < nNum:
215                     self.n[i] = nVec[j]
216                 else:
217                     self.n[i] = nVec[-1]
218                 j += 1
219             else:
220                 self.n[i] = nAir
221
222     ### functional method ###
223     if method == 'func':
224         for i in range(self.zNumFull):
225             if self.zFull[i] >= 0:
226                 if self.zFull[i] <= self.iceDepth:
227                     self.n[i] = nFunc(self.zFull[i])
228                 else:
229                     self.n[i] = nFunc(self.iceDepth)
230             else:
231                 self.n[i] = nAir
232
233     ### set reference index of refraction ###
234     self.n0 = self.at_depth(self.n, self.refDepth)
235
236     def get_n(self):
237         """
238         gets index of refraction profile of simulation

```

```

239
240     Returns
241     -----
242     1-d float array
243     """
244     return self.n[self.fNum:-self.fNum]
245
246 #Add 2D Refractive Index Profile -> Alex Kyriacou
247 def set_n2(self, method, nMat=None, nFunc=None, nAir=1.0003):
248     """
249         set the index of refraction profile of the simulation
250
251         future implementation plans:
252             - 2-d profiles
253             - complex index of refraction
254
255         Parameters
256         -----
257         method : string
258             'vector' for vector defined profile
259             'func' for function defined profile
260         nVec : array
261             if method=='vector', defines the index of refraction
262             profile of ice as an array
263             Precondition: spacing between elements is dz
264             Postcondition: n(z=0) = nVec[0], n(z=dz) = nVec[1], ... , n
265             (z>=len(nVec)*dz) = nVec[-1]
266         nMat : Array
267             if method == 'matrix' defines a matrix containing the
268             complex index of refracting profile of ice
269         nFunc : function
270             if method=='func', defines the index of refraction profile
271             of ice as a function
272             Precondition: nFunc is a function of one variable, z, and
273             returns a float value
274             Postcondition: n(z>=0) = nFunc(z)
275         nAir : float
276             index of refraction of air
277             Postcondition: n(z<0) = nAir
278     """
279     # self.n2 = np.ones((self.xNum, self.zNumFull), dtype='complex')
280     self.n2 = 1 + 1j * np.zeros((self.xNum, self.zNumFull))
281     ### vector method ###
282     if method == 'matrix':
283         for i in range(self.zNumFull):
284             self.n2[i, :] = nMat[i, :] # Note that the vector will have to include
285             the air surface as well
286
287     ### functional method ###
288     if method == 'func':
289         for i in range(self.xNum):
290             for j in range(self.zNumFull):

```

```

285         self.n2[i, j] = nFunc(self.x[i], self.zFull[j])
286
287     ### set reference index of refraction ###
288     self.n0 = self.at_depth(self.n, self.refDepth)
289     # self.n0 = self.at(self.n2, self.refDepth, self.refRange)
290
291     def get_n2(self):
292         """
293         gets index of refraction profile of simulation
294
295         Returns
296         -----
297         1-d float array
298         """
299         return self.n2
300
301     ### source functions ###
302     def set_user_source_profile(self, method, z0=0, sVec=None, sFunc=None):
303         """
304         set the spatial source profile explicitly (no frequency / signal information)
305         Precondition: index of refraction profile is already set
306
307         Parameters
308         -----
309         method : string
310             'vector' for vector defined profile
311             'func' for function defined profile
312         z0 : float
313             Precondition: z0>=0
314             reference starting point for sVec (m). Initialized to 0 m
315         sVec : array
316             if method=='vector', defines the source profile as an array
317             Precondition: spacing between elements is dz
318             Postcondition: E(z=z0) = sVec[0], E(z=z0+dz) = sVec[1], ... , E(z>=z0+len(
319 sVec)*dz) = sVec[-1], TODO
320         sFunc : function
321             if method=='func', defines the source profile as a function
322             Precondition: sFunc is a function of one variable, z, and returns a float
323 value
324             Postcondition: E(z>=0) = sFunc(z)
325         """
326         self.source = np.zeros(self.zNumFull, dtype='complex')
327
328     ### vector method ###
329     if method == 'vector':
330         sNum = len(sVec)
331         j = 0
332         for i in range(self.zNumFull):
333             if self.zFull[i] >= z0:
334                 if j < sNum:
335                     self.source[i] = sVec[j]
336                 else:

```

```

335         self.source[i] = 0
336         j += 1
337     else:
338         self.source[i] = 0
339
340     ### functional method ###
341     if method == 'func':
342         for i in range(self.zNumFull):
343             if self.zFull[i] >= 0:
344                 self.source[i] = sFunc(self.zFull[i])
345             else:
346                 self.source[i] = 0
347
348 def set_dipole_source_profile(self, centerFreq, depth, A=1+0.j):
349     """
350     set the source profile to be a half-wave dipole sized to center frequency
351     Precondition: index of refraction profile is already set
352
353     Parameters
354     -----
355     centerFreq : float
356         center frequency of to model dipole around (GHz)
357     depth : float
358         Precondition: depth>=0
359         depth of middle point of dipole (m)
360     A : complex float
361         complex amplitude of dipole. Initialized to 1 + 0j
362     """
363     ### frequency and wavelength in freespace ###
364     self.source = np.zeros(self.zNumFull, dtype='complex')
365     centerLmbda = util.c_light/centerFreq
366
367     ### wavelength at reference depth ###
368     centerLmbda0 = centerLmbda/self.n0
369
370     ### create dipole ###
371     z0 = depth
372     z0Index = util.findNearest(self.zFull, z0)
373
374     nPoints = int((centerLmbda0/2) / self.dz)
375     ZR1 = np.linspace(0,1, nPoints, dtype='complex')
376     ZR2 = np.linspace(1,0, nPoints, dtype='complex')
377     zRange = np.append(ZR1, ZR2)
378
379     n_x = np.pi*zRange
380     e = [0., 0., 1.]
381     beam = np.zeros(len(n_x), dtype='complex')
382     f0 = np.zeros(len(n_x), dtype='complex')
383
384     for i in range(len(n_x)):
385         n=[n_x[i], 0, 0]
386         val = np.cross(np.cross(n,e),n)[2]

```

```

387         beam[i] = complex(val, val)
388     f0 = A*(beam/(np.max(beam)))
389
390     self.source[z0Index-nPoints+1:z0Index+nPoints+1]=f0
391
392     def get_source_profile(self):
393         """
394         gets source profile of simulation
395
396         Returns
397         -----
398         1-d complex float array
399         """
400         return self.source[self.fNum:-self.fNum]
401
402
403     ### signal functions ###
404     def set_cw_source_signal(self, freq, amplitude = 1+0j):
405         """
406         set a continuous wave signal at a specified frequency
407
408         Parameters
409         -----
410         freq : float
411             frequency of source (GHz)
412         """
413         ### frequency ###
414         self.freq = np.array([freq], dtype='complex')
415         self.freqNum = len(self.freq)
416
417         ### wavenumber at reference depth ###
418         self.k0 = 2.*np.pi*self.freq*self.n0/util.c_light
419
420         ### coefficient ###
421         self.A = np.array([amplitude], dtype='complex')
422
423     def set_td_source_signal(self, sigVec, dt):
424         ### save input ###
425         self.dt = dt
426         self.sigVec = sigVec
427
428         ### frequencies ###
429         df = 1/(len(sigVec)*dt)
430         self.freq = np.arange(0, 1/dt, df, dtype='complex') #TODO -> Why do they
structure it like this?? why isn't the frequency space from -nyquist to +nyquist??
431         self.freqNum = len(self.freq)
432
433         ### wavenumbers at reference depth ###
434         self.k0 = 2.*np.pi*self.freq*self.n0/util.c_light
435
436         ### coefficient ###
437         self.A = util.doFFT(np.flip(sigVec))

```

```

438
439     # to ignore the DC component #
440     self.A[0] = self.k0[0] = 0
441
442
443     def get_spectrum(self):
444         """
445         gets transmitted signal spectrum
446
447         Returns
448         -----
449         1-d complex float array
450         """
451         return self.A[:int(self.freqNum/2)]
452
453     def get_frequency(self):
454         """
455         gets frequency array
456
457         Returns
458         -----
459         1-d float array
460         """
461         return abs(self.freq)[:int(self.freqNum/2)]
462
463     def get_signal(self):
464         """
465         gets transmitted signal
466
467         Returns
468         -----
469         1-d complex float array
470         """
471         return self.sigVec
472
473     def get_time(self):
474         """
475         gets time array
476
477         Returns
478         -----
479         1-d float array
480         """
481         return np.arange(0, self.dt*len(self.sigVec), self.dt)
482
483
484     ### field functions ###
485     def do_solver(self, rxList=np.array([])):
486         """
487         calculates field at points in the simulation
488         Precondition: index of refraction and source profiles are set
489

```

```

490     future implementation plans:
491         - different method options
492         - only store last range step option
493
494     Parameters
495     -----
496     rxList : array of Receiver objects
497         optional for cw signal simulation
498         required for non cw signal simulation
499     """
500
501     if (self.freqNum != 1):
502         ### check for Receivers ###
503         if (len(rxList) == 0):
504             print("Warning: Running time-domain simulation with no receivers. Field
505 will not be saved.")
506             for rx in rxList:
507                 rx.setup(self.freq, self.dt)
508
509         for j in np.arange(0, int(self.freqNum/2)+self.freqNum%2, 1, dtype='int'):
510             if (self.freq[j] == 0): continue
511             u = 2 * self.A[j] * self.source * self.filt * self.freq[j] #Set reduced
512 field u(0, z, f = f_j)
513             self.field[0,:] = u[self.fNum:-self.fNum] #Set Field at x=0 psi(x=0, z, f =
514 f_j)
515
516             ### method II ###
517             alpha = np.exp(1.j * self.dx * self.k0[j] * (np.sqrt(1. - (self.kz**2 /
518 self.k0[j]**2))- 1.))
519             B = (self.n)**2-1 #TODO Check if n is real or complex number -> does this
520 change the maths??
521             Y = np.sqrt(1.+(self.n/self.n0)**2)
522             beta = np.exp(1.j * self.dx * self.k0[j] * (np.sqrt(B+Y**2)-Y))
523
524             for i in range(1, self.xNum):
525                 u = alpha * (util.doFFT(u))
526                 u = beta * (util.doIFFT(u))
527                 u = self.filt * u
528
529                 self.field[i,:] = u[self.fNum:-self.fNum]/(np.sqrt(self.dx*i) * np.exp
530 (-1.j * self.k0[j] * self.dx * i))
531                 if (len(rxList) != 0):
532                     for rx in rxList:
533                         rx.add_spectrum_component(self.freq[j], self.get_field(x0=rx.x, z0=
534 rx.z))
535
536                 self.field.fill(0)
537
538     def do_solver2(self, rxList=np.array([]), freq_min = 0, freq_max = 1, nDiv = 1):
539         """
540         calculates field at points in the simulation
541         Precondition: index of refraction and source profiles are set
542 """

```

```

535     future implementation plans:
536
537         - different method options
538         - only store last range step option
539
540     Parameters
541     -----
542     rxList : array of Receiver objects
543             optional for cw signal simulation
544             required for non cw signal simulation
545     """
546     idx_min = util.findNearest(freq_min, self.freq)
547     idx_max = util.findNearest(freq_max, self.freq)
548
549     if (self.freqNum != 1):
550         ### check for Receivers ###
551         if (len(rxList) == 0):
552             print("Warning: Running time-domain simulation with no receivers. Field
will not be saved.")
553         for rx in rxList:
554             rx.setup(self.freq, self.dt)
555
556     #TODO: Add Sinc Interpolation:
557
558     #for k in range(1,self.freqNum):
559     for k in range(idx_min, idx_max):
560         if k % nDiv == 0:
561             tstart_k = time.time()
562             u = self.A[k] * self.source * self.filt
563             #print('u shape', u.shape)
564
565             self.field[0, :] = u[self.fNum:-self.fNum]
566             # print( u[self.fNum:-self.fNum].shape)
567             ### method II ###
568             # print(self.kz)
569
570             '''
571             alpha = np.exp(1.j * self.dx * self.k0[k] * (np.sqrt(1. - (self.kz ** 2
/ self.k0[k] ** 2)) - 1.))
572             B = (self.n2) ** 2 - 1
573             Y = np.sqrt(1. + (self.n2 / self.n0) ** 2)
574
575             beta = np.exp(1.j * self.dx * self.k0[k] * (np.sqrt(B + Y ** 2) - Y))
576             '''
577             range_times = []
578             for i in range(1, self.xNum):
579                 tstart_range = time.time()
580                 nVec = self.n2[i, :]
581                 # Added by Alex Kyriacou
582                 alpha = np.exp(1.j * self.dx * self.k0[k] * (
583                     np.sqrt(1. - (self.kz ** 2 / self.k0[k] ** 2)) - 1.))
584
585             # This is a 1D vector

```

```

584         B = nVec ** 2 - 1
585         Y = np.sqrt(1. + (nVec / self.n0) ** 2)
586         beta = np.exp(1.j * self.dx * self.k0[k] * (np.sqrt(B + Y ** 2) - Y
))
587
588         # TODO: Finish correcting this to be 2D -> fix tab
589         # u = alpha[i, :] * (util.doFFT(u))
590         u = alpha * (util.doFFT(u))
591         u = beta * (util.doIFFT(u))
592         u = self.filt * u
593
594         self.field[i, :] = u[self.fNum:-self.fNum] / (
595             np.sqrt(self.dx * i) * np.exp(-1.j * self.k0[k] * self.
dx * i))
596
597         # print(self.x[i], self.at_depth(self.field[i, :], 40))
598         tend_range = time.time()
599         range_times.append(tend_range-tstart_range)
600         range_time = np.mean(range_times)
601         if (len(rxList) != 0):
602             for rx in rxList:
603                 rx.add_spectrum_component(self.freq[k], self.get_field(x0=rx.x,
z0=rx.z))
604
605             # self.field.fill(0) #Deletes Field Afterwards
606             tend_k = time.time()
607             duration = tend_k - tstart_k
608             print('Solution complete, time: ', round(duration, 2), 's')
609             print('Average time per range step: ', range_time, ' approximate
solution for freq step: ', range_time*self.xNum)
610             nRemaining = (idx_max - k) / nDiv
611             print('Remaining Iterations', nRemaining)
612             remainder = datetime.timedelta(seconds=nRemaining * duration)
613             print('Remaining time: ' + str(remainder) + '\n')
614
615
616 def get_field(self, x0=None, z0=None):
617     """
618     gets field calculated by simulation
619
620     future implementation plans:
621     - interpolation option
622     - specify complex, absolute, real, or imaginary field
623
624     Parameters
625     -----
626     x0 : float
627         position of interest in x-dimension (m). optional
628     z0 : float
629         position of interest in z-dimension (m). optional
630
631     Returns
632     -----
633     if both x0 and z0 are supplied

```

```

632         complex float
633     if only one of x0 or z0 is supplied
634         1-d complex float array
635     if neither x0 or z0 are supplied
636         2-d complex float array
637     """
638     if (x0!=None and z0!=None):
639         return self.field[util.findNearest(self.x, x0),util.findNearest(self.z,z0)]
640     if (x0!=None and z0==None):
641         return self.field[util.findNearest(self.x, x0),:]
642     if (x0==None and z0!=None):
643         return self.field[:,util.findNearest(self.z,z0)]
644     return self.field
645
646
647     ### misc. functions ###
648     def at_depth(self, vec, depth):
649         """
650         find value of vector at specified depth.
651         future implementation plans:
652             - interpolation option
653             - 2D array searching. paraProp.at_depth() -> paraProp.at()
654
655         Parameters
656         -----
657         vec : array
658             vector of values
659             Precondition: len(vec) = len(z)
660         depth : float
661             depth of interest (m)
662
663         Returns
664         -----
665         base type of vec
666         """
667         ### error if depth is out of bounds of simulation ###
668         if (depth > self.iceDepth or depth < -self.airHeight):
669             print("Error: Looking at z-position of out bounds")
670             return np.NaN
671
672         # find closest index #
673         dIndex = round((depth + self.filterDepth + self.airHeight) / self.dz)
674
675         return vec[dIndex]
676
677     def backwards_solver(self, rxList= np.array([]), freq_min = 0, freq_max = 0, nDiv=
1, R_threshold = 0.1):
678         # Cut over your frequency space
679         #freq_cut = util.cut_xaxis(self.freq, freq_min, freq_max)
680
681         if self.freqNum > 1:
682             idx_min = util.findNearest(freq_min, self.freq)

```

```

683         idx_max = util.findNearest(freq_max, self.freq)
684     else:
685         idx_min = 0
686         idx_max = 1
687         #nFreq = len(freq_cut)
688
689         ### check for Receivers ###
690         if (len(rxList) == 0):
691             print("Warning: Running time-domain selfulation with no receivers. Field
will not be saved.")
692         for rx in rxList:
693             rx.setup(self.freq, self.dt)
694
695         for iFreq in range(idx_min, idx_max):
696             if iFreq % nDiv == 0:
697                 tstart_i = time.time() # Start a time for every frequency step
698                 freq_i = self.freq[iFreq] # Frequency_i
699
700                 print('solving for: f = ', freq_i, 'GHz, A = ', self.A[iFreq], 'step:',
iFreq - idx_min, 'steps left:', idx_max - iFreq)
701
702                 # Add U_positive field
703                 u_plus = 2 * self.A[iFreq] * self.source * self.filt * freq_i # Set
Forward Propogating Field u_plus
704                 self.field[0, :] = u_plus[self.fNum:-self.fNum]
705                 alpha = np.exp(1.j * self.dx * self.k0[iFreq] * (np.sqrt(1. - (self.kz/
self.k0[iFreq])**2) - 1.))
706
707                 for jXstep in range(1, self.xNum):
708                     n_j = self.n2[jXstep, :] # Verticle referactive index profile at
x_i, n_i(z) = n(x = x_i, z)
709                     x_j = self.x[jXstep]
710
711                     B = n_j ** 2 - 1
712                     Y = np.sqrt(1. + (n_j / self.n0) ** 2)
713                     beta = np.exp(1.j * self.dx * self.k0[iFreq] * (np.sqrt(B + Y ** 2)
- Y))
714
715                     u_plus = alpha * (util.doFFT(u_plus))
716                     u_plus = beta * (util.doIFFT(u_plus))
717                     u_plus = self.filt * u_plus
718
719                     self.field[jXstep, :] = (u_plus[self.fNum:-self.fNum] / np.sqrt(x_j
)) * np.exp(1.j * self.k0[iFreq] * x_j)
720
721                     #Calculate Reflections range-wise
722                     dNx = n_j - self.n2[jXstep - 1, :]
723                     reflelction_z = util.reflection_coefficient(n_j, self.n2[jXstep -
1, :]) # Calculate Reflection coefficient
724
725                     if any(reflelction_z) > R_threshold: #TODO -> Should this be
reflection_z ** 2?

```

```

726         refl_field = np.zeros((jXstep+1, self.zNum), dtype='complex')
727
728         refl_source = np.zeros(self.zNumFull, dtype='complex')
729         refl_source[self.fNum:-self.fNum] = self.field[jXstep, :] *
reflelction_z[self.fNum:-self.fNum]
730         #Scale forward going reduced field by transmission coefficient
(has to be smaller than last one)
731         u_plus[self.fNum:-self.fNum] *= util.transmission_coefficient(
n_j, self.n2[jXstep - 1, :])[self.fNum:-self.fNum] #TODO -> field or u_plus??
732
733         #Create Negative Travelling Reduced Field
734         print(len(refl_source), len(self.filt) )
735         u_minus = 2 * refl_source * self.filt * self.freq[iFreq]
736         refl_field[jXstep, :] = u_minus[self.fNum:-self.fNum]
737
738         nSteps_backwards = jXstep - 1 # Number of Steps back to origin
x=0
739         mXstep = jXstep - 1 # This indice goes backwards
740
741         alpha_minus = np.exp(1.j * self.dx * self.k0[iFreq] * (np.sqrt
(1. - (self.kz/ self.k0[iFreq])**2) - 1.))
742         for kBack in range(nSteps_backwards):
743             n_k = self.n2[mXstep, :]
744             x_minus = abs(self.x[mXstep] - x_j)
745
746             B_minus = n_k ** 2 - 1
747             Y_minus = np.sqrt(1. + (n_k / self.n0) ** 2)
748             beta_minus = np.exp(1.j * self.dx * self.k0[iFreq] * (np.
sqrt(B_minus + Y_minus ** 2) - Y_minus))
749
750             u_minus = alpha_minus * (util.doFFT(u_minus))
751             u_minus = beta_minus * (util.doIFFT(u_minus))
752             u_minus = self.filt * u_minus
753
754             refl_field[mXstep, :] = (u_minus[self.fNum:-self.fNum] / np
.sqrt(x_minus)) * np.exp(1j*x_minus*self.k0[iFreq])
755             mXstep -= 1
756
757             self.field[:,jXstep, :] += refl_field[:,jXstep, :])#TODO -> Should
I be adding field components or reduced field components??
758             if len(rxList) > 0:
759                 for rx in rxList:
760                     rx.add_spectrum_component(self.freq[iFreq], self.get_field(x0=
rx.x, z0=rx.z))
761             tend_i = time.time()
762             duration = tend_i - tstart_i
763             print('Solution complete, time: ', round(duration,2), 's')
764             nRemaining = (idx_max-iFreq)/nDiv
765             print('Remaining Iterations', nRemaining)
766             remainder = datetime.timedelta(seconds = nRemaining*duration)
767             print('Remaining time: ' + str(remainder) + '\n')
768

```

```

769     def backwards_solver_2way(self, rxList = np.array([]), freq_min = 0, freq_max = 1,
770     nDiv=1, R_threshold=0.1):
771         #New method for calculating backwards waves using u_minus -> use a 3D array to
772         hold reflection sources
773         # Cut over your frequency space
774         # freq_cut = util.cut_xaxis(self.freq, freq_min, freq_max)
775         if self.freqNum > 1:
776             idx_min = util.findNearest(freq_min, self.freq)
777             idx_max = util.findNearest(freq_max, self.freq)
778         else:
779             idx_min = 0
780             idx_max = 1
781         # nFreq = len(freq_cut)
782
783         ### check for Receivers ###
784         if (len(rxList) == 0):
785             print("Warning: Running time-domain selfulation with no receivers. Field
786             will not be saved.")
787         for rx in rxList:
788             rx.setup(self.freq, self.dt)
789         for iFreq in range(idx_min, idx_max):
790             if iFreq % nDiv == 0:
791                 tstart_i = time.time() # Start a time for every frequency step
792                 freq_i = self.freq[iFreq] # Frequency_i
793
794                 print('solving for: f = ', freq_i, 'GHz, A = ', self.A[iFreq], 'step:',
795                 iFreq - idx_min, 'steps left:', idx_max - iFreq)
796
797                 # Add U_positive field
798                 u_plus = 2 * self.A[iFreq] * self.source * self.filt * freq_i # Set
799                 Forward Propogating Field u_plus
800                 self.field[0, :] = u_plus[self.fNum:-self.fNum]
801                 alpha = np.exp(1.j * self.dx * self.k0[iFreq] * (np.sqrt(1. - (self.kz/
802                 self.k0[iFreq])**2) - 1.))
803
804                 #Backwards Reflection Source
805                 #refl_source_3arr = np.zeros((self.xNum, self.zNum), dtype='complex')
806                 refl_source_list = [] #list that contains the reflection sources
807                 nRefl = 0
808
809                 #Solve for u_plus -> from x = 0, x = R
810                 time_plus_l = []
811                 time_minus_l = []
812                 time_xtotal_l = []
813                 tstart_i = time.time()
814                 for jXstep in range(1, self.xNum):
815                     tstart_xplus = time.time()
816
817                     n_j = self.n2[jXstep, :] # Verticle referactive index profile at
818                     x_i, n_i(z) = n(x = x_i, z)
819                     x_j = self.x[jXstep]

```

```

814         B = n_j ** 2 - 1
815         Y = np.sqrt(1. + (n_j / self.n0) ** 2)
816         beta = np.exp(1.j * self.dx * self.k0[iFreq] * (np.sqrt(B + Y ** 2)
- Y))
817
818         u_plus = alpha * (util.doFFT(u_plus))
819         u_plus = beta * (util.doIFFT(u_plus))
820         u_plus = self.filt * u_plus
821
822         self.field[jXstep, :] = (u_plus[self.fNum:-self.fNum] / np.sqrt(x_j
)) * np.exp(1.j * self.k0[iFreq] * x_j)
823         tend_xplus = time.time()
824         time_plus_l.append(tend_xplus - tstart_xplus)
825
826         # Calculate Reflections range-wise
827         dNx = n_j - self.n2[jXstep - 1, :]
828         reflelction_z = util.reflection_coefficient(n_j, self.n2[jXstep -
1,:]) # Calculate Reflection coefficient
829
830         if any(reflelction_z) > R_threshold:
831             nRefl += 1
832             refl_source = np.zeros(self.zNumFull, dtype='complex')
833             refl_source[self.fNum:-self.fNum] = self.field[jXstep, :] *
reflelction_z[self.fNum:-self.fNum]
834
835             refl_field = np.zeros((self.xNum, self.zNum), dtype='complex')
836             refl_field[jXstep,:] = refl_source[self.fNum:-self.fNum]
837             refl_source_list.append(refl_field)
838             # Scale forward going reduced field by transmission coefficient
(has to be smaller than last one)
839             u_plus[self.fNum:-self.fNum] *= util.transmission_coefficient(
n_j, self.n2[jXstep - 1, :])[self.fNum:-self.fNum] # TODO -> field or u_plus??
840
841             #Complete forward propagation
842             #Commence backwards propagation
843             if nRefl > 0:
844                 refl_source_3arr = np.zeros((self.xNum, self.zNumFull, nRefl),
dtype='complex')
845                 for k in range(nRefl):
846                     refl_source_3arr[:,self.fNum:-self.fNum,k] = refl_source_list[k
]
847
848                 mXstep = self.xNum - 1
849                 u_minus = np.zeros((nRefl, self.zNumFull), dtype='complex')
850                 alpha_minus = np.exp(1.j * self.dx * self.k0[iFreq] * (np.sqrt(1. -
(self.kz / self.k0[iFreq]) ** 2) - 1.))
851                 refl_field_3arr = np.zeros((self.xNum, self.zNum, nRefl))
852
853                 print('zNum',self.zNum, 'zNumFull (including filtered depths', self
.zNumFull)
854                 for kBack in range(1, self.xNum): #Make j steps backwards
855                     tstart_xminus = time.time()

```

```

856         if refl_source_3arr[kBack].any() > 0:
857             filt2 = np.array([self.filt]*nRefl)
858             u_minus[:, :] += 2 * np.transpose(refl_source_3arr[mXstep
, :, :]) * filt2 * self.freq[iFreq]
859
860             n_k = np.array([self.n2[mXstep, :]]*nRefl)
861             x_minus = abs(self.iceLength - self.x[mXstep])
862
863             B_minus = n_k ** 2 - 1
864             #n0_k = np.array([self.n0]*nRefl)
865             Y_minus = np.sqrt(1. + (n_k / self.n0) ** 2)
866             k0_k = np.array([self.k0]*nRefl)
867             beta_minus = np.exp(1.j * self.dx * k0_k * (np.sqrt(B_minus +
Y_minus ** 2) - Y_minus))
868
869             filt_k = np.array([self.filt]*nRefl)
870
871             #TODO: Check if FFT can operate on a 2D array?
872             u_minus = alpha_minus * (util.doFFT(u_minus))
873             #print(u_minus.shape)
874             u_minus = beta_minus * (util.doIFFT(u_minus))
875             #print(u_minus.shape)
876             u_minus = filt_k * u_minus
877             #print(u_minus.shape)
878
879             mXstep -= 1
880             refl_field_3arr[mXstep, :, :] = np.transpose((u_minus[:, self.
fNum:-self.fNum] / np.sqrt(x_minus)) * np.exp(1j * x_minus * k0_k))
881             tend_xminus = time.time()
882             time_minus_l.append(tend_xminus - tstart_xminus)
883         for k in range(nRefl):
884             self.field[:, :] += refl_field_3arr[:, :, k]
885             tend_i = time.time()
886             print('time per pos x step', np.mean(time_plus_l))
887             print('time per negative x step', np.mean(time_minus_l))
888             print('simulation per frequency step', tend_i - tstart_i, 'time per x
step (average)', (tend_i-tstart_i)/self.xNum)
889             if len(rxList) > 0:
890                 for rx in rxList:
891                     rx.add_spectrum_component(self.freq[iFreq], self.get_field(x0=
rx.x, z0=rx.z))

```

## Run-Simulation.py

```

1 #!/usr/bin/env python
2 # coding: utf-8
3
4 # In[1]:
5
6
7 # paraPropPython example use of paraPropPython.py notebook

```

```

8 # s. prohira, c. sbrocco
9
10 #get_ipython().run_line_magic('matplotlib', 'inline')
11 import paraPropPython as ppp
12 import numpy as np
13 import matplotlib.pyplot as pl
14 import scipy
15 from scipy import signal
16
17
18 # In[2]:
19
20
21 from paralell_scan import *
22 from pulse import *
23
24
25 # In[3]:
26
27
28 ### an example of defining n as a function of z (also can be done using a vector, see
    implementation) ###
29 def southpole(z):
30     A=1.78
31     B=-0.43
32     C=-0.0132
33     return A+B*np.exp(C*z)
34
35 def enceladus_pole(z, snow_depth = 10, firn_depth = 20, ice_depth = 100):
36     n_ice = 1.82
37     n_firn = np.sqrt(2.2)
38     n_snow = np.sqrt(1.2)
39     n_water = np.sqrt(82.)
40     n0 = 1.0003
41
42     nz = n_water
43     if z < 0:
44         nz = n0
45     elif z >= 0 and z < snow_depth:
46         nz = n_snow
47     elif z >= snow_depth and z < firn_depth:
48         nz = n_firn
49     elif z >= firn_depth and z < ice_depth:
50         nz = n_ice
51     return nz
52
53 def vacuum(z):
54     if z > 0:
55         nz = 1.0003
56     return 1.0003
57
58

```

```
59 def pure_ice(z):
60     epsilon_ice_1 = 3.2
61     epsilon_ice_2 = 1e-5
62     nz = np.sqrt(epsilon_ice_1 + 1j * epsilon_ice_2)
63     return nz
64
65 def pure_ice_air(z):
66     epsilon_ice_1 = 3.2
67     epsilon_ice_2 = 1e-5
68     n_ice = np.sqrt(epsilon_ice_1 + 1j * epsilon_ice_2)
69     n_vac = 1
70
71     if z < 0:
72         nz = n_vac
73     elif z >= 0:
74         nz = n_ice
75     return nz
76
77
78 def pure_ice_water_pocket(z, ice_depth_1=50, water_pocket_depth=150, ice_depth_2=200):
79     epsilon_ice_1 = 3.2
80     epsilon_ice_2 = 1e-5
81     epsilon_water_1 = 74
82     epsilon_water_2 = 1.8
83     n_water = np.sqrt(82.)
84     n0 = 1
85     nz = n_water
86     if z < 0:
87         nz = n0
88     elif z >= 0 and z < ice_depth_1:
89         nz = np.sqrt(epsilon_ice_1 + 1j * epsilon_ice_2)
90     elif z >= ice_depth_1 and z < water_pocket_depth:
91         nz = np.sqrt(epsilon_water_1 + 1j * epsilon_water_2)
92     elif z >= water_pocket_depth and z < ice_depth_2:
93         nz = np.sqrt(epsilon_ice_1 + 1j * epsilon_ice_2)
94     return nz
95
96 def enceladus_reference(z, snow_depth = 25):
97     epsilon_ice_1 = 3.19
98     epsilon_ice_2 = 1e-5
99     n_ice = np.sqrt(epsilon_ice_1 + 1j * epsilon_ice_2)
100     epsilon_snow_1 = 1.2
101     epsilon_snow_2 = 0.139
102     n_snow = np.sqrt(epsilon_snow_1 + 1j * epsilon_snow_2)
103     n0 = 1
104
105
106     if z < 0:
107         nz = n0
108     elif z >= 0 and z < snow_depth:
109         nz = n_snow
110     elif z >= snow_depth:
```

```

111     nz = n_ice
112     return nz
113
114 def enceladus_water_pocket_new(z, snow_depth = 25, ice_depth_1 = 50, water_pocket_depth
    = 150, ice_depth_2 = 200):
115     epsilon_ice_1 = 3.19
116     epsilon_ice_2 = 1e-5
117     n_ice = np.sqrt(epsilon_ice_1 + 1j * epsilon_ice_2)
118     epsilon_snow_1 = 1.2
119     epsilon_snow_2 = 0.139
120     n_snow = np.sqrt(epsilon_snow_1 + 1j * epsilon_snow_2)
121     epsilon_water_1 = 85.355
122     epsilon_water_2 = 90.884
123     n_water = np.sqrt(epsilon_water_1 + 1j * epsilon_water_2)
124     n0 = 1
125
126
127     if z < 0:
128         nz = n0
129     elif z >= 0 and z < snow_depth:
130         nz = n_snow
131     elif z >= snow_depth and z < ice_depth_1:
132         nz = n_ice
133     elif z >= ice_depth_1 and z < water_pocket_depth:
134         nz = n_water
135     elif z >= water_pocket_depth and z < ice_depth_2:
136         nz = n_ice
137
138     return nz
139
140
141
142
143 def enceladus_water_pocket(z, snow_depth=-20, firn_depth=-10, surface_depth = 0 ,
    ice_depth_1 = 50, water_pocket_depth = 150, ice_depth_2 = 200):
144     epsilon_ice_1 = 3.2
145     epsilon_ice_2 = 1e-5
146     epsilon_water_1 = 74
147     epsilon_water_2 = 1.8
148     n_ice = np.sqrt(epsilon_ice_1 + 1j * epsilon_ice_2)
149     n_water = np.sqrt(epsilon_water_1 + 1j * epsilon_water_2)
150     n_firn = np.sqrt(2.2)
151     n_snow = np.sqrt(1.2)
152     #epsilon_snow_1 =
153     #epsilon_snow_2 =
154     n_snow = np.sqrt(epsilon_snow_1 + 1j * epsilon_snow_2)
155     n0 = 1
156     nz = n_water
157
158     if z < snow_depth:
159         nz = n0
160     elif z >= snow_depth and z < firn_depth:

```

```

161     nz = n_snow
162     elif z >= firn_depth and z < surface_depth:
163         nz = n_firn
164     elif z >= surface_depth and z < ice_depth_1:
165         nz = n_ice
166     elif z >= ice_depth_1 and z < water_pocket_depth:
167         nz = n_water
168     elif z >= water_pocket_depth and z < ice_depth_2:
169         nz = n_ice
170
171     return nz
172
173 # In[4]:
174
175
176 #pulse_file = path_to_directory + simul_directory + 'pulse_input.npy'
177 #np.save(pulse_file, pulse_input)
178 ### first, initialize an instance of paraProp by defining its dimensions and frequency
    of interest ###
179 from scipy import signal
180
181
182 #sim = ppp.paraProp(iceLength, iceDepth, dx, dz, refDepth=sourceDepth)
183
184
185 frequency_central = 500e6 #Frequency of Transmitter Pulse
186 t_end = 1000e-9 #Time length of pulse [s]
187 t_start = 0 #starting time
188 t_samples = 1/2e9 #Sample Interval
189 amplitude0 = 1 #TX Amplitude
190
191 t_pulse = 200e-9 #Time of Pulse centre
192
193
194 tx_pulse = pulse(amplitude0, frequency_central, t_start, t_end, t_samples) #Start TX
    Pulse object
195
196 #tx_pulse.do_gaussian(t_pulse) #Set TX pulse to gaussian pulse
197 tcentral = t_pulse
198 i, q, e = signal.gausspulse(tx_pulse.time_space - tcentral, fc=tx_pulse.frequency,
    retquad=True, retenv=True)
199 tx_pulse.real = i
200 tx_pulse.imag = q
201 tx_pulse.abs = e
202 tx_pulse.centre = tcentral
203
204 nSamples = tx_pulse.nSamples
205
206 fig = pl.figure(figsize=(12,12), dpi = 100)
207 ax1 = fig.add_subplot(211)
208 ax2 = fig.add_subplot(212)
209

```

```

210 ax1.set_title('Pulse (time space)')
211 ax1.plot(tx_pulse.time_space*1e9, tx_pulse.real, label='real')
212 ax1.plot(tx_pulse.time_space*1e9, tx_pulse.imag, label='imag')
213 ax1.plot(tx_pulse.time_space*1e9, tx_pulse.abs, label='abs')
214
215 ax1.legend()
216 ax1.grid()
217 ax1.set_xlabel('Time [ns]')
218 ax1.set_ylabel('Amplitude [V]')
219
220 ax2.set_title('Pulse FFT (freq space)')
221 ax2.plot(tx_pulse.freq_space/1e6, abs(tx_pulse.doFFT()), label='abs')
222 ax2.plot(tx_pulse.freq_space/1e6, tx_pulse.doFFT().real, label='real')
223 ax2.plot(tx_pulse.freq_space/1e6, tx_pulse.doFFT().imag, label='imag')
224 ax2.set_xlabel('Frequency [MHz]')
225 ax2.set_ylabel('Amplitude [V]')
226
227 ax2.grid()
228 ax2.legend()
229 pl.show()
230
231 # In[5]:
232
233 iceDepth = 150. # m
234 iceLength = 120. # m
235 dx = 1 # m
236 dz = 0.05 # m
237
238 ### it is useful to set the reference depth as the source depth when you only have one
    transmitter ###
239 sourceDepth = 50. # m
240 sim = ppp.paraProp(iceLength, iceDepth, dx, dz, refDepth=sourceDepth)
241 z = sim.get_z()
242 x = sim.get_x()
243
244 #Define Receiver Positions:
245 max_depth_tx = 50.
246 max_depth_rx = 50.
247 step_interval = 2
248 nDepths_rx = int(max_depth_rx / step_interval) + 1
249
250 nTransmitters = nDepths_rx
251 sourceDepth_list = np.linspace(0, max_depth_tx, nTransmitters)
252 receiverDepth_list = np.linspace(0, max_depth_rx, nDepths_rx)
253
254 #Range List #Change Range of Receivers
255 Range_list = np.array( [10, 20, 30, 50, 80, 100] )
256 nRanges = len(Range_list)
257
258 rx_array = np.ones((nDepths_rx, nRanges, 2))
259 rx_list = []
260 for i in range(nDepths_rx):

```

```

261     for j in range(nRanges):
262         rx_array[i, j][0] = Range_list[j]
263         rx_array[i, j][1] = receiverDepth_list[i]
264         rx_list.append([Range_list[j], receiverDepth_list[i]])
265
266 fig = pl.figure(figsize=(15,8), dpi = 100)
267 ax1 = fig.add_subplot(121)
268 ax2 = fig.add_subplot(122)
269
270 ax1.set_title('Transmitter and \n Receiver Positions', fontsize=25)
271 ax1.scatter(np.array(rx_list)[:,:0], -np.array(rx_list)[:,:1], label='RX')
272
273 tx_array = np.ones(nTransmitters)
274
275 for i in range(nTransmitters):
276     tx_array[i] = sourceDepth_list[i]
277 ax1.scatter(np.zeros(nTransmitters), -tx_array, label='TX')
278 ax1.legend(shadow=True, loc='upper right', fontsize=17)
279 ax1.grid()
280 ax1.set_ylabel('z [m]', fontsize=25)
281 ax1.set_xlabel('x [m]', fontsize=25)
282 ax2.set_xlabel('Refractive Index n', fontsize=20)
283
284
285 #Load a profile from vector
286 #alpline_profile = np.genfromtxt('share/alpine-schumman-indOfRef_5cm.dat')
287 #sim.set_n('vector', nVec = alpline_profile[:,1])
288 #sim.set_n('func', nFunc=enceladus_pole)
289
290 method = 'func'
291 profile = 'enceladus_water_pocket_new'
292 sim.set_n(method, nFunc=enceladus_water_pocket_new)
293 ### plot ###
294
295
296 ax2.plot(sim.get_n(), -z, color='black')
297 ax2.grid()
298 ax2.set_title('Index of Refraction Profile \n of Simulation', fontsize=25)
299 ax2.text(1.2, 15, 'Vacuum', fontsize = 20, family= 'serif')
300 ax2.text(1.5, -12, 'Impure Snow', fontsize = 20, family = 'serif')
301 ax2.text(2.2, -35, 'Pure Ice', fontsize = 20, family = 'serif')
302 ax2.text(3.9, -70, 'Saline Water', fontsize = 20, family = 'serif')
303 ax2.set_ylim(-max_depth_rx - 35, 35)
304 ax1.set_ylim(-max_depth_rx - 5, 13)
305 ax1.tick_params(labelsize=20)
306 ax2.tick_params(labelsize=20)
307 plt.show()
308
309
310 pl.show()
311
312

```

```

313 sim.set_dipole_source_profile(frequency_central/1e9, max_depth_rx/2)
314 sim.set_cw_source_signal(frequency_central/1e9)
315
316
317 ### run the solver ###
318 sim.do_solver()
319
320 ### plot absolute value of field for whole simulation space ###
321 fig = pl.figure(figsize=(15,8), dpi = 100)
322 ax = fig.add_subplot(111)
323
324 pmesh = pl.imshow(np.transpose(abs(sim.get_field())), aspect='auto', cmap='hot', vmin
    =1e-5, vmax=5e-2,
325     extent=(x[0], x[-1], z[-1], z[0]))
326 cbar = pl.colorbar(pmesh)
327 pl.title("Absolute Field, " + str(int((frequency_central/1e9)*1000))+" MHz", fontsize
    =25)
328 pl.xlabel("x (m)", fontsize=25)
329 pl.ylabel("z (m)", fontsize=25)
330 pl.xticks(fontsize=20)
331 pl.yticks(fontsize=20)
332 pl.show()
333
334
335 # In[6]:
336
337
338 from numpy.lib.format import open_memmap
339
340 def create_memmap(fname, path_to_directory, shape0, dtype0 = 'float32', mode0 = 'w+'): #
    Create a Blank Memmap for recording data
341     full_file = path_to_directory + fname
342     A = open_memmap(full_file, dtype=dtype0, mode=mode0, shape=shape0)
343     return A
344
345 def generate_ascan(pulse_tx, tx_array, rx_array, path_to_file, file_name): #Creates an
    Array of Scans for a set of receivers and transmitters
346     nRanges_rx = len(rx_array[0])
347     nDepths_rx = len(rx_array)
348     nDepths_tx = len(tx_array)
349
350     freq_space = pulse_tx.freq_space
351     nFreqs = len(freq_space)
352
353     fft_pulse = np.fft.fft(pulse_tx.real)
354
355     ascan_array = create_memmap(file_name, path_to_file, shape0 = (nFreqs, nDepths_tx,
    nRanges_rx, nDepths_rx), dtype0=complex)
356     #full_file = path_to_file + file_name
357     print(nFreqs, nDepths_tx, nDepths_rx, nRanges_rx)
358     #ascan_array = open_memmap(full_file, dtype=complex, mode='w+', shape=(nFreqs,
    nDepths_tx, nRanges_rx, nDepths_rx))

```

```

359
360     for i in range(1, len(freq_space)):
361         print(round(float(i)/len(freq_space)*100,2), '%, frequency = ', int(freq_space[
i]/1e6), 'MHz')
362         frequency_i = freq_space[i]
363         amplitude_fft = fft_pulse[i]
364
365         for j in range(nDepths_tx):
366             #for j in range(1): # Here!!!
367                 sourceDepth = tx_array[j]
368                 print('source depth: ', sourceDepth)
369
370                 sim.set_dipole_source_profile(abs(frequency_i) / 1e9, sourceDepth,
fft_pulse[i]) # Set the dipole source
371                 sim.set_cw_source_signal(abs(frequency_i) / 1e9) # Set the frequency
372                 sim.do_solver() # Run the simulation
373
374                 for k in range(nRanges_rx):
375                     #for k in range(1):
376                         for l in range(nDepths_rx):
377                             #for l in range(1):
378
379                                 RX = rc(rx_array[l,k][0], rx_array[l,k][1]) #Get Receiver position
(range and depth)
380                                 print(round(float(i)/len(freq_space)*100,2), '%')
381                                 print('RX: ', RX.x, RX.z, 'source depth: ', sourceDepth, 'frequency
= ', int(freq_space[i]/1e6), 'MHz')
382
383                                 signal_rx = sim.get_field(RX.x, RX.z)
384                                 print(ascan_array.shape)
385                                 ascan_array[i, j, k, l] = signal_rx
386
387                                 print('TX amplitude: ', 10*np.log10(abs(fft_pulse[i])), 'RX
amplitude: ', 10*np.log10(abs(signal_rx)))
388                                 print('')
389                                 #np.save(rx_array)
390                                 #np.save(tx_array)
391                                 #np.save(freq_space)
392
393                 return ascan_array, tx_array, rx_array
394
395 # In[11]:
396
397 #CREATE THE FOLDER WHERE YOUR SIMULATION TAKES PLACE
398 path_to_file = 'test/' #You can Rename the folder when you want to make a new
simulation!!
399 if not os.path.exists(path_to_file):
400     os.mkdir(path_to_file)
401
402 #Create the file name containing your received pulses
403 file_name = 'ascan.npy'
404 full_path = path_to_file + file_name

```

```

405
406 #Create the meta data files
407 np.save(path_to_file + 'freq_space.npy', tx_pulse.freq_space) #Frequencies used in
    Simulation
408 np.save(path_to_file + 'time_space.npy', tx_pulse.time_space) #Time space of pulse
409 np.save(path_to_file + 'tx_pulse.npy', tx_pulse.real + 1j*tx_pulse.imag) #Transmitter
    pulse (complex)
410 np.save(path_to_file + 'tx_array.npy', tx_array) #Array of Transmitter positions (1D)
411 np.save(path_to_file + 'rx_array.npy', rx_array) #Array of Receiver positions (2D)
412
413 #Save Simulation Data to External File
414 info_file = path_to_file + 'simul_info.txt'
415 fout = open(info_file, 'w+')
416
417 fout.write(method + '\t#method\n')
418 fout.write(profile + '\tnProfile\n')
419 fout.write(str(frequency_central/1e6) + '\t#Central-Frequency-MHz\n')
420 fout.write(str(tx_pulse.nSamples) + '\t#Number-Samples\n')
421 fout.write(str(t_samples) + '\t#Sampling-interval-s\n')
422 nyquist_frequency = 1/(2*t_samples)
423 fout.write(str(nyquist_frequency)+'\t#nyquist-freauency-Hz\n')
424
425 now = datetime.now()
426 date_str = now.strftime('%Y.%m.%d.%H:%M:%S')
427 fout.write(date_str + '\t#datetime\n')
428 fout.close()
429
430 # In[ ]:
431
432 #Run your simulation -> Results are saved to 'file_name' under
433 generate_ascan(tx_pulse, tx_array, rx_array, path_to_file, file_name)

```

# E

## Bibliography

- O Abramov, D Raggio, PM Schenk, and JR Spencer. Temperatures of vents within enceladus' tiger stripes. In *Lunar and Planetary Science Conference*, number 1832, page 1206, 2015.
- Antonin Affholder, François Guyot, Boris Sauterey, Régis Ferrière, and Stéphane Mazevet. Bayesian analysis of Enceladus's plume data to assess methanogenesis. *Nature Astronomy*, pages 1–10, 2021. doi: 10.1038/s41550-021-01372-6.
- AH Andersen and AC Kak. Digital ray tracing in two-dimensional refractive fields. *The journal of the Acoustical society of America*, 72(5):1593–1606, 1982.
- Gokhan Apaydin and Levent Sevgi. *Radio wave propagation and parabolic equation modeling*. John Wiley & Sons, 2017.
- PRF Barnes, Eric W Wolff, Robert Mulvaney, R Udisti, E Castellano, R Röthlisberger, and J-P Stefensen. Effect of density on electrical conductivity of chemically laden polar ice. *Journal of Geophysical Research: Solid Earth*, 107(B2):ESE–1, 2002.
- Stephen Battersby. Saturn's moon Enceladus surprisingly comet-like. <https://www.newscientist.com/article/dn13541-saturns-moon-enceladus-surprisingly-comet-like/>, 2008. Online; Last Retrieved: 06-07-2021.
- Vitalii Vasilevich Bogorodsky, Charles R Bentley, and PE Gudmandsen. *Radioglaciology*, volume 1. Springer Science & Business Media, 2012.
- Robert H Brown, Roger N Clark, Bonnie J Buratti, Dale P Cruikshank, Jason W Barnes, Rachel ME Mastrapa, J Bauer, S Newman, T Momary, KH Baines, et al. Composition and physical properties of Enceladus' surface. *Science*, 311(5766):1425–1428, 2006.
- BJ Buratti, PA Dalba, MD Hicks, and JP Chang. Enceladus: Surface Texture and Roughness as Clues to What Lies Beneath. In *Lunar and Planetary Science Conference*, number 1777, page 2038, 2014.
- CJF Böttcher. Theory of electric polarization, Ch. 2, 1952.
- J. Castillo, D. Matson, C. Sotin, T. Johnson, J. Lunine, and P. Thomas. A New Understanding of the Internal Evolution of Saturnian Icy Satellites from Cassini Observations. 2006.
- J. C. Castillo, D. L. Matson, T. V. Johnson, J. I. Lunine, T. B. McCord, C. Sotin, P. C. Thomas, and E. B. Turtle. 26A1 in the Saturnian System - New Interior Models for the Saturnian satellites. In *AGU Fall Meeting Abstracts*, volume 2005, pages P32A–01, December 2005.

- 
- Gaël Choblet, Gabriel Tobie, Christophe Sotin, Marie Běhouňková, Ondřej Čadek, Frank Postberg, and Ondřej Souček. Powering prolonged hydrothermal activity inside Enceladus. *Nature Astronomy*, 1(12):841–847, 2017.
- Christopher F Chyba, Steven J Ostro, and Bradley C Edwards. Radar detectability of a subsurface ocean on Europa. *Icarus*, 134(2):292–302, 1998.
- Kenneth S Cole and Robert H Cole. Dispersion and absorption in dielectrics I. Alternating current characteristics. *The Journal of chemical physics*, 9(4):341–351, 1941.
- comsol.de. Dipole Antenna with a Quarter-Wave Coaxial Balun. <https://www.comsol.de/model/dipole-antenna-with-a-quarter-wave-coaxial-balun-12313>. Last Retrieved: 09-20-2021.
- D Craig Jones and Nick Hudyma. Characterization of Weathered Limestone Using Cross-Hole Ground Penetrating Radar Techniques. In *Geo-Chicago 2016*, pages 475–486.
- David J Daniels. Surface-penetrating radar. *Electronics & Communication Engineering Journal*, 1996.
- Wim Degruyter and M Manga. Cryoclastic origin of particles on the surface of enceladus. *Geophysical research letters*, 38(16), 2011.
- DLR. Home: Programme Missionen: Navigation: Projekte: Die EnEx-Initiative. [https://www.dlr.de/rd/desktopdefault.aspx/tabid-10572/18379\\_read-42824/](https://www.dlr.de/rd/desktopdefault.aspx/tabid-10572/18379_read-42824/). Last Retrieved: 09-20-2021.
- Renée Dotson. *Enceladus and the Icy Moons of Saturn*. University of Arizona Press, 2018. ISBN 9780816537075. URL <http://www.jstor.org/stable/j.ctv65sw2b>.
- Michele Dougherty, Larry Esposito, and Stamatios Krimigis. *Saturn from Cassini-huygens*. Springer Science & Business Media, 2009.
- DC Dube. Study of Landau-Lifshitz-Looyenga’s formula for dielectric correlation between powder and bulk. *Journal of Physics D: Applied Physics*, 3(11):1648, 1970.
- Michael Efroimsky. Tidal viscosity of Enceladus. *Icarus*, 300:223–226, 2018.
- Jonas Ehn. Propagation of radio waves in a realistic environment using a parabolic equation approach, 2019.
- Dmitry Eliseev. Entwicklung Elektronischer Systeme für die Akustische Navigation einer Manövrierbaren Schmelzsonde im Eis. *Diss. RWTH Aachen*, pages 3–6, 2018.
- W Ellison, A Balana, G Delbos, K Lamkaouchi, L Eymard, C Guillou, and C Prigent. New permittivity measurements of seawater. *Radio science*, 33(3):639–648, 1998.
- William Ellison. Permittivity of Pure Water, at Standard Atmospheric Pressure, over the Frequency Range 0–25 THz and the Temperature Range 0–100 °C. *Journal of Physical and Chemical Reference Data*, 36:1, 03 2007. doi: 10.1063/1.2360986.
- S Evans. Dielectric properties of ice and snow—a review. *Journal of Glaciology*, 5(42):773–792, 1965.

- MD Feit and JA Fleck. Light propagation in graded-index optical fibers. *Applied optics*, 17(24):3990–3998, 1978.
- P Friend and A Kyriacou. Dielectric Profiling of Enceladus’ Surface in the Tiger Stripe Region. In *Lunar and Planetary Science Conference*, number 2548, page 2586, 2021.
- Randall German. *Sintering: from empirical observations to scientific principles*. Butterworth-Heinemann, 2014.
- Christopher R Glein, John A Baross, and J Hunter Waite Jr. The pH of Enceladus’ ocean. *Geochimica et Cosmochimica Acta*, 162:202–219, 2015.
- JW Glen. The effect of hydrogen disorder on dislocation movement and plastic deformation of ice. *Physik der kondensierten Materie*, 7(1):43–51, 1968.
- JW Glen and JG Paren. The electrical properties of snow and ice. *Journal of Glaciology*, 15(73):15–38, 1975.
- Xiao-Wei Guan, Li-Xin Guo, Ya-Jiao Wang, and Qing-Liang Li. Parabolic Equation Modeling of Propagation over Terrain Using Digital Elevation Model. *International Journal of Antennas and Propagation*, 2018, 2018.
- B Gundlach, J Ratte, J Blum, J Oesert, and SN Gorb. Sintering and sublimation of micrometre-sized water-ice particles: The formation of surface crusts on icy Solar System bodies. *Monthly Notices of the Royal Astronomical Society*, 479(4):5272–5287, 2018.
- RH Hardin and F Tappert. Applications of the split-step Fourier method to the numerical solution of nonlinear and variable coefficient wave equations. *SIAM Review (Chronicles)*, 15, 1973.
- Alain Hérique, Benoit Agnus, E Asphaug, A Barucci, Pierre Beck, Julie Bellerose, Jens Biele, Lydie Bonal, Pierre Bousquet, Lorenzo Bruzzone, et al. Direct observations of asteroid interior and regolith structure: science measurement requirements. *Advances in Space Research*, 62(8):2141–2162, 2018.
- PD. Dr. Christian Hoelbling. Elektrodynamik und spezielle Relativitätstheorie. [https://moodle.uni-wuppertal.de/pluginfile.php/908664/mod\\_resource/content/49/tp2.pdf](https://moodle.uni-wuppertal.de/pluginfile.php/908664/mod_resource/content/49/tp2.pdf), 2020. Online; last downloaded: 2021-05-19.
- Claude Jaccard. Thermodynamics of irreversible processes applied to ice. *Physik der kondensierten Materie*, 3(2):99–118, 1964.
- R Jaumann, K Stephan, GB Hansen, RN Clark, BJ Buratti, RH Brown, KH Baines, SF Newman, G Bellucci, G Filacchione, et al. Distribution of icy particles across enceladus’ surface as derived from cassini-vims measurements. *Icarus*, 193(2):407–419, 2008.
- Wanying Kang, Tushar Mittal, Suyash Bire, Jean-Michel Campin, and John Marshall. How does salinity shape ocean circulation and ice geometry on Enceladus and other icy satellites? *arXiv preprint arXiv:2104.07008*, 2021.
- Evangelia A Karagianni. Electromagnetic waves under sea: bow-tie antennas design for wi-fi underwater communications. *Progress In Electromagnetics Research M*, 41:189–198, 2015.

- 
- Konstantinos Konstantinidis, Claudio L Flores Martinez, Bernd Dachwald, Andreas Ohndorf, Paul Dykta, Pascal Bowitz, Martin Rudolph, Ilya Digel, Julia Kowalski, Konstantin Voigt, et al. A lander mission to probe subglacial water on Saturn's moon Enceladus for life. *Acta astronautica*, 106:63–89, 2015.
- Alexander Luke Kyriacou. Enceladus Explorer: Radar based sounding system and navigation for a melting probe to investigate the ocean of Enceladus for signs of life, 2018. URL <https://indico.ecap.work/event/1/contributions/65/attachments/39/92/astroparticle-school-enex.pdf>. Last Retrieved: 09-20-2021.
- LD Landau and EM Lifshitz. *Electrodynamics of Continuous Media*, Perga, 1960.
- MF Levy and AA Zaporozhets. Parabolic equation techniques for scattering. *Wave Motion*, 31(2): 147–156, 2000.
- H Looyenga. Dielectric constants of heterogeneous mixtures. *Physica*, 31(3):401–406, 1965.
- Dennis L Matson, Julie C Castillo-Rogez, Ashley Gerard Davies, and Torrence V Johnson. Enceladus: A hypothesis for bringing both heat and chemicals to the surface. *Icarus*, 221(1):53–62, 2012.
- DL Matson, JC Castillo, C Sotin, TV Johnson, JI Lunine, AG Davies, TB McCord, PC Thomas, and EP Turtle. Enceladus' Interior and Geysers-Possibility for Hydrothermal Geochemistry and N<sub>2</sub> Production. In *37th Annual Lunar and Planetary Science Conference*, pages abstract–no. League City, Texas, 2006.
- Christopher P McKay, Carolyn C Porco, Travis Altheide, Wanda L Davis, and Timothy A Kral. The possible origin and persistence of life on Enceladus and detection of biomarkers in the plume. *Astrobiology*, 8(5):909–919, 2008.
- Jamie L Molaro, Mathieu Choukroun, Cynthia B Phillips, Eli S Phelps, Robert Hodyss, Karl L Mitchell, Juan M Lora, and Gareth Meirion-Griffith. The microstructural evolution of water ice in the solar system through sintering. *Journal of Geophysical Research: Planets*, 124(2):243–277, 2019.
- John C Moore. Models of radar absorption in European ice. *Icarus*, 147(1):292–300, 2000.
- John C Moore, Robert Mulvaney, and JG Paren. Dielectric stratigraphy of ice: A new technique for determining total ionic concentrations in polar ice cores. *Geophysical Research Letters*, 16(10): 1177–1180, 1989.
- Carl D Murray and Stanley F Dermott. *Solar system dynamics*. Cambridge university press, 1999.
- NASA. Enceladus: Ocean Moon. <https://solarsystem.nasa.gov/missions/cassini/science/enceladus/>, 2018. Online; Last Retrieved: 06-07-2021.
- NASA/JPL. Photojournal: Saturn: Saturn's Rings (Artist Concept). <https://photojournal.jpl.nasa.gov/catalog/PIA03550>, 2005. Last Retrieved: 09-20-2021.
- NASA/JPL. Photojournal: Saturn: Trapping of Methane In Enceladus' Ocean. <https://photojournal.jpl.nasa.gov/catalog/PIA19059>, 2015. Last Retrieved: 09-20-2021.

- Ernest H Nickel and Joel D Grice. The IMA Commission on New Minerals and Mineral Names: procedures and guidelines on mineral nomenclature, 1998. *Mineralogy and Petrology*, 64(1):237–263, 1998.
- Wolfgang Nolting. Elektrodynamik. In *Grundkurs Theoretische Physik 3*, pages 221–435. Springer, 2013.
- Elena Pettinelli, Barbara Cosciotti, Federico Di Paolo, Sebastian Emanuel Lauro, Elisabetta Mattei, Roberto Orosei, and Giuliano Vannaroni. Dielectric properties of Jovian satellite ice analogs for subsurface radar exploration: A review. *Reviews of Geophysics*, 53(3):593–641, 2015.
- Carolyn Porco, Daiana DiNino, and Francis Nimmo. How the geysers, tidal stresses, and thermal emission across the south polar terrain of Enceladus are related. *The Astronomical Journal*, 148(3):45, 2014.
- Carolyn C Porco, Paul Helfenstein, PC Thomas, AP Ingersoll, J Wisdom, Robert West, Gerhard Neukum, Tilmann Denk, Roland Wagner, Thomas Roatsch, et al. Cassini observes the active south pole of Enceladus. *science*, 311(5766):1393–1401, 2006.
- Frank Postberg, Sascha Kempf, Jürgen Schmidt, N Brilliantov, Alexander Beinsen, Bernd Abel, Udo Buck, and Ralf Srama. Sodium salts in E-ring ice grains from an ocean below the surface of Enceladus. *Nature*, 459(7250):1098–1101, 2009.
- Frank Postberg, Roger N Clark, Candice J Hansen, Andrew J Coates, Cristina M Dalle Ore, Francesca Scipioni, M Hedman, and J Waite. Plume and surface composition of Enceladus. *Enceladus and the icy moons of Saturn*, pages 129–162, 2018.
- S Prohira, C Sbrocco, P Allison, J Beatty, D Besson, A Connolly, P Dasgupta, C Deaconu, KD de Vries, et al. Modeling in-ice radio propagation with parabolic equation methods. *Physical Review D*, 103(10):103007, 2021.
- John Raven, Ken Caldeira, Harry Elderfield, Ove Hoegh-Guldberg, Peter Liss, Ulf Riebesell, John Shepherd, Carol Turley, and Andrew Watson. *Ocean acidification due to increasing atmospheric carbon dioxide*. The Royal Society, 2005.
- F Scipioni, P Schenk, Federico Tosi, EMILIANO D’Aversa, R Clark, J-Ph Combe, and CM Dalle Ore. Deciphering sub-micron ice particles on enceladus surface. *Icarus*, 290:183–200, 2017.
- S Shabtaie and CR Bentley. Unified theory of electrical conduction in firn and ice: Site percolation and conduction in snow and firn. *Journal of Geophysical Research: Solid Earth*, 99(B10):19757–19769, 1994.
- Evert Slob, Motoyuki Sato, and Gary Olhoeft. Surface and borehole ground-penetrating-radar developments. *Geophysics*, 75(5):75A103–75A120, 2010.
- Joshua Sokol. Icy Enceladus’s tiger stripes are a window on its watery depths. <https://www.newscientist.com/article/2128414-icy-enceladuss-tiger-stripes-are-a-window-on-its-watery-depths/>, 2017. Online; Last Retrieved: 09-06-2021.

- 
- Ben Southworth, Sascha Kempf, and Joe Spitale. Surface Deposition of the Enceladus Plume and the Angle of Emissions. *Icarus*, 319, 01 2018. doi: 10.1016/j.icarus.2018.08.024.
- Ben S Southworth, Sascha Kempf, and Joe Spitale. Surface deposition of the Enceladus plume and the zenith angle of emissions. *Icarus*, 319:33–42, 2019.
- GH Spencer and MVRK Murty. General ray-tracing procedure. *JOSA*, 52(6):672–678, 1962.
- John R Spencer and Francis Nimmo. Enceladus: An active ice world in the Saturn system. *Annual Review of Earth and Planetary Sciences*, 41:693–717, 2013.
- Joseph N Spitale and Carolyn C Porco. Association of the jets of Enceladus with the warmest regions on its south-polar fractures. *Nature*, 449(7163):695–697, 2007.
- Joseph N Spitale, Terry A Hurford, Alyssa R Rhoden, Emily E Berkson, and Symeon S Platts. Curtain eruptions from Enceladus’ south-polar terrain. *Nature*, 521(7550):57–60, 2015.
- Dennis M Sullivan. *Electromagnetic simulation using the FDTD method*. John Wiley & Sons, 2013.
- J Hunter Waite, Christopher R Glein, Rebecca S Perryman, Ben D Teolis, Brian A Magee, Greg Miller, Jacob Grimes, Mark E Perry, Kelly E Miller, Alexis Bouquet, et al. Cassini finds molecular hydrogen in the Enceladus plume: evidence for hydrothermal processes. *Science*, 356(6334):155–159, 2017.
- J Hunter Waite Jr, WS Lewis, BA Magee, JI Lunine, WB McKinnon, CR Glein, O Mousis, DT Young, T Brockwell, J Westlake, et al. Liquid water on Enceladus from observations of ammonia and 40 Ar in the plume. *Nature*, 460(7254):487–490, 2009.
- Catherine C Walker, Michael W Liemohn, and Christopher D Parkinson. On radar sounding applications for Enceladean ice. In *2010 IEEE International Geoscience and Remote Sensing Symposium*, pages 4522–4525. IEEE, 2010.
- wikipedia.org. Cole-Cole-Diagramm der dielektrischen Funktion von Wasser bei 0°C. [https://www.dlr.de/rd/desktopdefault.aspx/tabid-10572/18379\\_read-42824/](https://www.dlr.de/rd/desktopdefault.aspx/tabid-10572/18379_read-42824/), a. Last Retrieved: 09-20-2021.
- wikipedia.org. Physik Oberstufe/ Schwingungen und Wellen/ Elektromagnetische Wellen. [https://de.wikibooks.org/wiki/Physik\\_Oberstufe/\\_Schwingungen\\_und\\_Wellen/\\_Elektromagnetische\\_Wellen](https://de.wikibooks.org/wiki/Physik_Oberstufe/_Schwingungen_und_Wellen/_Elektromagnetische_Wellen), b. Last Retrieved: 09-21-2021.
- wikipedia.org. Raytrace changing refractive index. 2008. Last Retrieved: 09-20-2021.
- wikipedia.org. FDTD Yee grid 2d-3d. [https://commons.wikimedia.org/wiki/File:FDTD\\_Yee\\_grid\\_2d-3d.svg](https://commons.wikimedia.org/wiki/File:FDTD_Yee_grid_2d-3d.svg), 2015. Last Retrieved: 09-20-2021.
- Eric W Wolff, William D Miners, John C Moore, and Julian G Paren. Factors controlling the electrical conductivity of ice from the polar regions A summary. *The Journal of Physical Chemistry B*, 101(32):6090–6094, 1997.
- Kane Yee. Numerical solution of initial boundary value problems involving Maxwell’s equations in isotropic media. *IEEE Transactions on antennas and propagation*, 14(3):302–307, 1966.

# Acknowledgements

I would like to take this opportunity to thank all the people who supported me in writing this Bachelor thesis.

First of all, I would like to duly thank *Prof. Dr. Klaus Helbing* for giving me the opportunity to write the thesis in his working group. I would also like to thank *Prof. Dr. Christopher Wiebusch* for being the co-referee of my thesis.

The completion of this thesis could not have been realized without the great help of *Alex Kyriacou*. My deepest and sincere gratitude for patiently helping me with programming and teaching me a lot of things, whether it was data analysis or scientific writing. In general, thanks to all my colleagues from room F.11.02 who always had an open ear for me and supported me significantly during the work. Thank you, *Dr. Pia Friend*, for all the interesting and informative conversations we had, as well as for proofreading my thesis and maybe also for being picky about the correction, as it has helped me enormously to understand factual contexts correctly. Thank you, *Fabian Becker*, for all the pleasant conversations in the office and for advise and assistance every time.

This thesis, but also the time during the entire course of study, would have been much more nerve-racking if I hadn't had such good friends around me, starting off with my fellow students *Max, Patrick* and *Niklas*. Without you, the time would only have been half as fun and bearable. Special thanks to friends from outside the university go also to *Marvin, Samuel, Max, Daniel, Alex* and *Maurice*, who have accompanied me on this path.

Un doveroso ringraziamento va ovviamente alla mia famiglia, perché mi è stata accanto e non mi ha fatta mai mancare il suo sostegno e il suo aiuto durante questi anni, senza la quale non avrei mai neppur cominciare questo cammino e non sarei dove sono ora, raggiungendo questo immenso obiettivo. Consegno anche a lei virtualmente questo diploma di laurea in segno di riconoscimento per gli sforzi sostenuti, non solo finanziari, ma anche considerando tutti i sacrifici che hanno fatto per questo, allontanandosi per così tanto tempo dalla famiglia in Italia. Per questo sono eternamente grato. Grazie a te, mamma *Josiane*, per avermi sempre aiutato a superare i momenti più difficili, soprattutto confortandomi e incoraggiandomi nel periodo degli esami perché credevo di non farcela. Grazie a te, papà *Antonio*, per avermi insegnato le cose importanti nella vita e istruendomi, che mi hanno fatto crescere e diventare un adulto e permettendomi di arrivare fin qui oggi. E grazie a te, sorella *Maria*, nonostante che litigavamo non di rado, potevo sempre confidarmi con te e spero che tu possa raggiungere tutti i tuoi obiettivi e desideri.

# Declaration of Authorship / Eidesstattliche Erklärung

according to §15(6) of the examination regulations of 11/25/2019  
gem. §15 Abs. 6 der Prüfungsordnung vom 25.11.2019

Surname (Name), First name (Vorname): Boccarella, Gianluca

## English

I hereby declare in lieu of oath that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references.

This thesis was not previously presented to another examination board and has not been published.

I hereby also agree that my thesis (Bachelor thesis) may be made available for inspection by academically interested persons or institutions and within the framework of external quality assurance measures of the degree programme. Correction or evaluation notes in my thesis may not be quoted.

## Deutsch

Hiermit versichere ich an Eides Statt, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben, sowie Zitate kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Hiermit erkläre ich mich auch damit einverstanden, dass meine Abschlussarbeit (Bachelor-Thesis) wissenschaftlich interessierten Personen oder Institutionen und im Rahmen von externen Qualitätssicherungsmaßnahmen des Studienganges zur Einsichtnahme zur Verfügung gestellt werden kann. Korrektur- oder Bewertungshinweise in meiner Arbeit dürfen nicht zitiert werden.

Wuppertal, den \_\_\_\_\_

\_\_\_\_\_  
(Gianluca Boccarella)